

Обробка зображень і мультимедіа

Олег Гутік



Лекція 20: Стиснення зображень, XIV. JPEG

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибрати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настроюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

JPEG є дуже витонченим методом стиснення зображень із втратою та без втрати інформації. Він застосовується як до кольорових, так і до напівтонових зображень (але не до мультфільмів та анімацій). Цей алгоритм не дуже добре стискає дворівневі чорно-білі образи, але він чудово обробляє зображення з неперервними тонами, в яких близькі пікселі мають схожі кольори. Важливою перевагою методу JPEG є велика кількість параметрів, що настраюються, які користувач може вибирати на свій розсуд, зокрема, він може регулювати відсоток втрати інформації, а, отже, і коефіцієнт стиснення, в широкому діапазоні. Зазвичай око не в змозі помітити будь-якої шкоди навіть при стисканні цим методом у 10 або 20 разів. Є два основні режими: з втратою (який називається також *базеліною*) і без втрат інформації (який не надто ефективний та зазвичай дає фактор стиснення близько 2). Більшість додатків насамперед використовують режим із втратою даних. Цей режим також містить прогресуюче та ієрархічне кодування.

JPEG є насамперед методом стиснення. Його не можна розглядати як повноцінний стандарт представлення зображень. Тому в ньому не задаються такі специфічні параметри зображення як геометричний розмір пікселя, простір кольорів або чергування рядків біта.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1. Високий коефіцієнт стиснення, особливо для зображень, які складаються з великої кількості шкрявочок та шпигачок.
- 2. Велика кількість параметрів, що дозволяють досягнути оптимального співвідношення між стиснутим методом та досягати необхідного балансу стиснення/якості.
- 3. Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх складності, простоти форм, кольору тонових відтінків та контрастності.
- 4. Достатньо витончений метод стиснення, що не вимагає складної та повільної стиснення/відновлення програм, та проста програма реалізації методу для більшості стандартних платформ.
- 5. Декілька режимів роботи.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1. Високий коефіцієнт стиснення, який дозволяє зберігати зображення в файлах розміром до 100 разів менше за вихідні.
- 2. Велика кількість параметрів, що дозволяють досягнути певного компромісу між якістю зображення з використанням методу та довжиною збереженого балансу стиснення/якості.
- 3. Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх складності, контрасту, простору кольорів, розміру файлів та інших властивостей.
- 4. Достатньо витончений метод стиснення, що набагато швидше працює, ніж деякі старіші методи, які використовувалися раніше, та надає програмі реалізації методу для того, щоб зберегти достатню швидкість.
- 5. Декілька режимів роботи.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1. Високий коефіцієнт стиснення, який дозволяє зберігати зображення в форматі файлів розміром до 100 разів менше за оригінал.
- 2. Велика кількість параметрів, що дозволяють досягнути певного компромісу між якістю зображення і компактністю методу та дозволяють знайти оптимальний баланс стиснення і якості.
- 3. Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх складності, контрасту, кольору, розміру, масштабу та інших властивостей.
- 4. Достатньо витончений метод стиснення, що на відміну від інших методів дозволяє створювати відносно компактні та швидко програмує реалізацію методу для більшості сучасних платформ.
- 5. Декілька режимів роботи.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - послідовний режим
 - прогресивний режим
 - режим без втрат
 - ієрархічний режим

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Стиснення зображень. JPEG

JPEG був розроблений як метод стиснення безперервно-тонових образів. Основні цілі методу JPEG полягають у наступному:

- 1 Високий коефіцієнт стиснення, особливо для зображень, якість яких розцінюється як хороша чи відмінна.
- 2 Велика кількість параметрів, що дозволяють досвідченому користувачеві експериментувати з налаштуваннями методу та домагатися необхідного балансу стиснення/якість.
- 3 Хороші результати для будь-яких типів неперервно-тонових зображень незалежно від їх роздільної здатності, простору кольорів, розміру пікселів або інших властивостей.
- 4 Достатньо витончений метод стиснення, але не надто складний, що дозволяє створювати відповідні пристрої та писати програми реалізації методу для комп'ютерів більшості платформ.
- 5 Декілька режимів роботи:
 - 1 **послідовний режим**: кожен колірний компонент стискається в одному скануванні зліва направо, зверху вниз;
 - 2 **прогресивний режим**: зображення стискається в кілька блоків (відомих як "скани"), щоб розпакувати та переглянути від грубих до дрібних деталей;
 - 3 **режим без втрат**: важливий для випадків, коли користувач вирішує, що пікселі не повинні бути втрачені (компромісом є низький коефіцієнт стиснення порівняно з режимами з втратами);
 - 4 **ієрархічний режим**: зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропущується для зображень у градациях сірого. Цей крок можна додати після етапу яскравості, до того як закінчити тому частину алгоритму, що стосується кольору, в обох випадках і тактичними. Бути кращим рішенням. Об'єктом цього кроку є перетворення двох зображень з RGB на яскравість/кольор. Для цього використовують матрицю перетворення з чотирьох компонентів кольору в яскравість/кольор. Цей перетворення простору кольорів у простір яскравості/кольору можна додати до етапу яскравості/кольору, до того як закінчити етап доведення зображень до монохромного вигляду.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропущується для зображень у градівній шкалі. Цей крок можна додати між кроками 2 та 3, якщо потрібно, тому частіше відомий як крок 2.5. Він не виконується в усіх реалізаціях, але є важливою частиною стандарту. Крім того, він виконується для зображень у форматі YCbCr, який є основним форматом для зображень у форматі JPEG. Цей крок виконується за допомогою матриці перетворення кольору, яка перетворює простір кольорів RGB на простір яскравості/кольору. Цей крок виконується за допомогою матриці перетворення кольору, яка перетворює простір кольорів RGB на простір яскравості/кольору. Цей крок виконується за допомогою матриці перетворення кольору, яка перетворює простір кольорів RGB на простір яскравості/кольору.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропущується для зображень у градівній шкалі. Цей крок може допомогти зменшити кількість даних, які використовуються, тому частенько використовується між двома етапами стиснення. Також можна бути певно впевненими, що об'єкту якого-небудь кольору, який займає більшу частину зображення, буде достатньо описати його в яскравості/кольорі, а не в RGB. Крім того, якщо зображення складається з великої кількості кольорів, то його можна описати в яскравості/кольорі, а не в RGB. Крім того, якщо зображення складається з великої кількості кольорів, то його можна описати в яскравості/кольорі, а не в RGB.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок потрібен тільки для зображень з трьома каналами. Для чорно-білих зображень цей крок не потрібний. Після цього кроку зображення складається з двох частин: кольорової частини (якщо вона є) та каналу яскравості. Ці частини обробляються окремо. Після цього кроку зображення перетворюється з RGB на канал яскравості/кольору. Цей крок потрібен тільки для зображень з трьома каналами. Після цього кроку зображення складається з двох частин: кольорової частини (якщо вона є) та каналу яскравості. Ці частини обробляються окремо. Після цього кроку зображення перетворюється з RGB на канал яскравості/кольору. Цей крок потрібен тільки для зображень з трьома каналами. Після цього кроку зображення складається з двох частин: кольорової частини (якщо вона є) та каналу яскравості. Ці частини обробляються окремо.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок проводиться для всіх зображень у форматі зображення. Цей крок виконується для кожного пікселя окремо, до того як закартовити зображення в матрицю. Після цього зображення зберігається в форматі YCbCr. Цей крок виконується для кожного пікселя окремо, до того як закартовити зображення в матрицю. Після цього зображення зберігається в форматі YCbCr. Цей крок виконується для кожного пікселя окремо, до того як закартовити зображення в матрицю. Після цього зображення зберігається в форматі YCbCr.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок проводиться для всіх зображень у форматі зображення з кольором. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
2. Зображення розділяється на блоки розміром 8x8 пікселів. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
3. Для кожного блоку 8x8 пікселів обчислюються коефіцієнти дискретного косинусного перетворення (ДКП). Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
4. Коефіцієнти ДКП сортуються за зростаючим значенням абсолютної величини. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
5. Сортовані коефіцієнти ДКП кодуються за допомогою кодування з нульовими значеннями (ZCV). Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
6. Кодовані коефіцієнти ДКП збираються в єдиний біт-стрім. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
7. Біт-стрім збирається в єдиний біт-стрім. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.
8. Біт-стрім збирається в єдиний біт-стрім. Цей крок виконується для кожного каналу кольору окремо, що дозволяє використовувати менше біт на канал кольору.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ІСО (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору;

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору;

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль CCITT та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

Термін JPEG є аббревіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ISO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках.

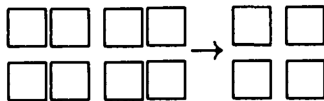
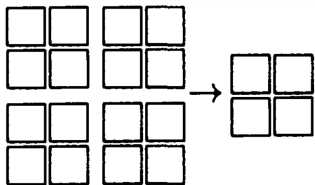
Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

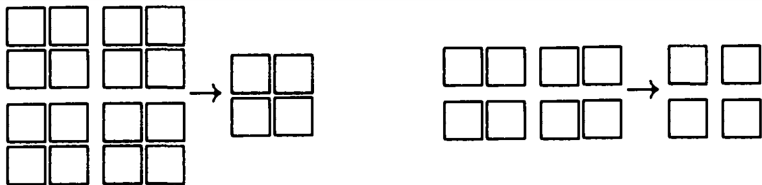
Термін JPEG є абрєвіатурою, яка розшифровується як *Joint Photographic Experts Group*. Проект JPEG був результатом спільних зусиль ССІТТ та ІSO (the International Standard Organization, Міжнародна організація стандартів), який розпочався в червні 1987 року та підготував перший проект JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках. Нижче наведено основні кроки стиснення JPEG, а кожен крок докладно описано пізніше.

1. Кольорові зображення перетворюються з RGB на колірний простір яскравості/кольору; цей крок пропускається для зображень у градаціях сірого). Око чутливе до невеликих змін яскравості, але не до кольоровості, тому частини кольоровості можуть згодом втратити багато даних і, таким чином, бути сильно стиснутими без значного погіршення загальної якості зображення. Цей крок необов'язковий, але важливий, оскільки решта алгоритму працює з кожним компонентом кольору окремо. Без перетворення простору кольорів із компонентів RGB не можна видалити істотну частину інформації, що не дозволяє зробити сильний стиск.

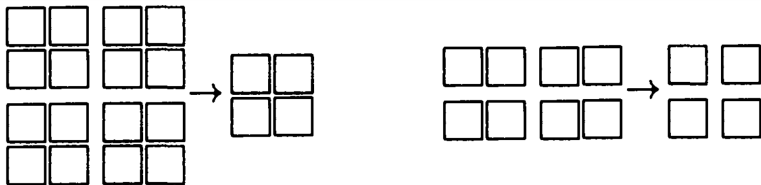
Стиснення зображень. JPEG



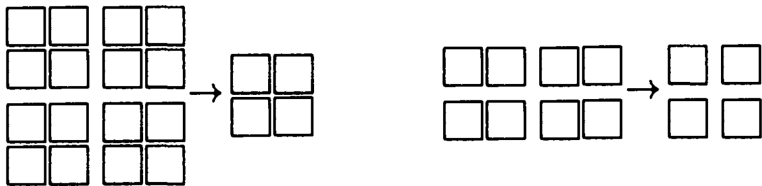
- Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних. Цей крок зменшує розмір файлу, який потрібно зберегти, і дозволяє зберегти лише дані, необхідні для зображення у графічному форматі. Зменшення роздільної здатності виконується для кожного каналу кольору окремо. Зменшення роздільної здатності виконується шляхом створення пікселів із низькою роздільною здатністю з вихідних. Цей крок зменшує розмір файлу, який потрібно зберегти, і дозволяє зберегти лише дані, необхідні для зображення у графічному форматі. Зменшення роздільної здатності виконується шляхом створення пікселів із низькою роздільною здатністю з вихідних. Цей крок зменшує розмір файлу, який потрібно зберегти, і дозволяє зберегти лише дані, необхідні для зображення у графічному форматі.
- Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних. Цей крок зменшує розмір файлу, який потрібно зберегти, і дозволяє зберегти лише дані, необхідні для зображення у графічному форматі. Зменшення роздільної здатності виконується шляхом створення пікселів із низькою роздільною здатністю з вихідних. Цей крок зменшує розмір файлу, який потрібно зберегти, і дозволяє зберегти лише дані, необхідні для зображення у графічному форматі.



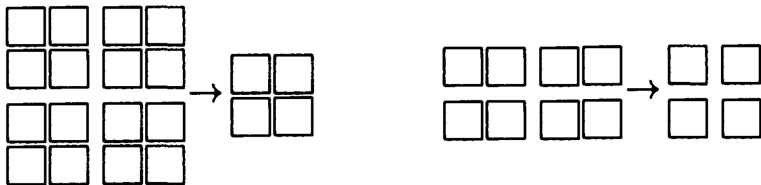
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



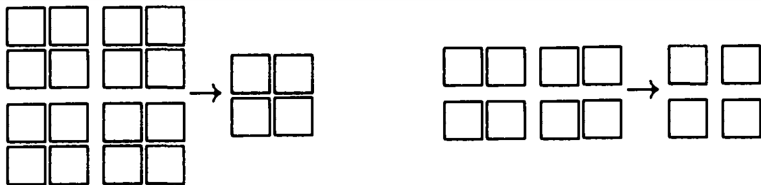
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



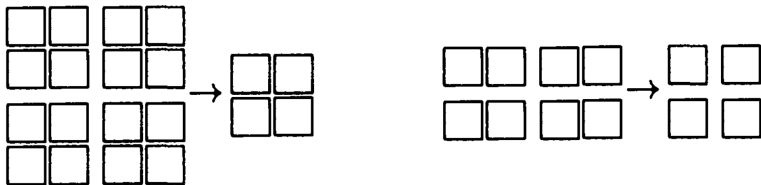
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



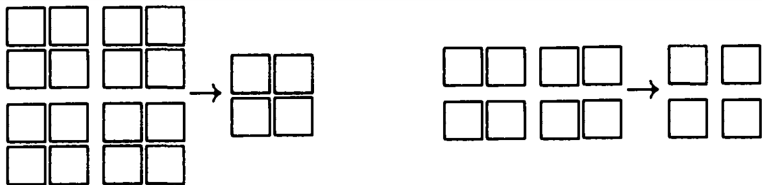
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



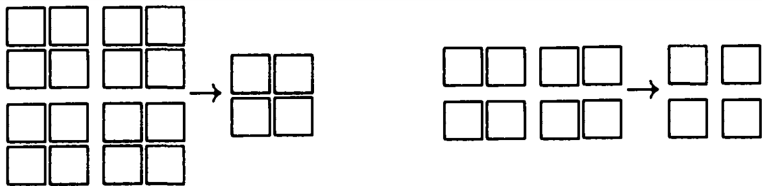
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



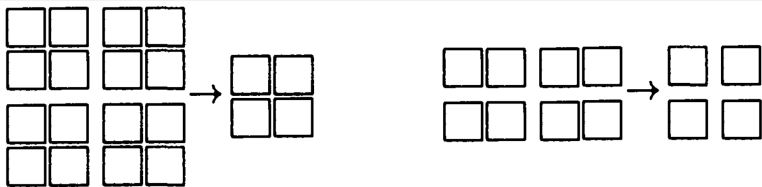
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



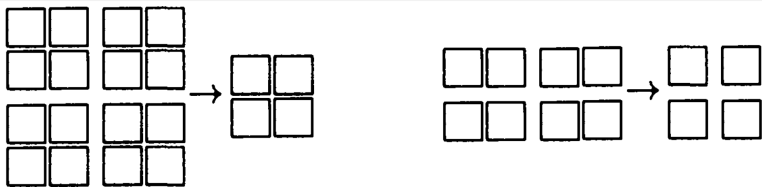
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



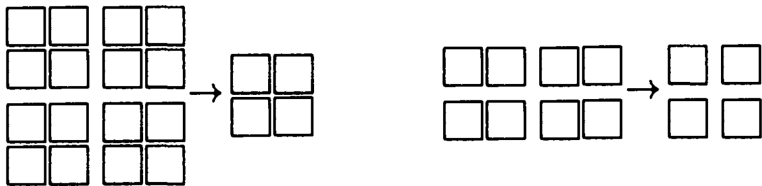
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



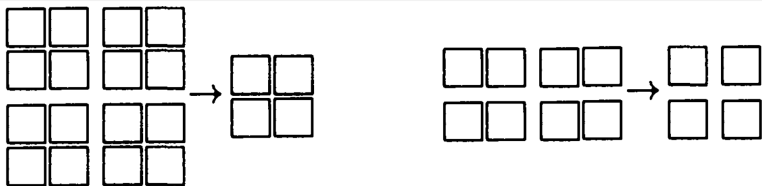
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



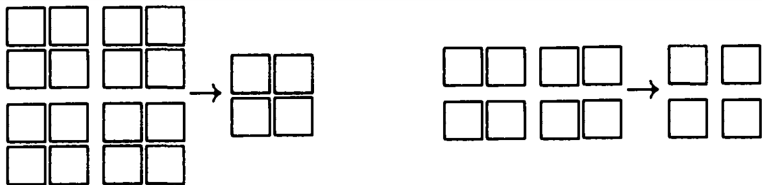
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



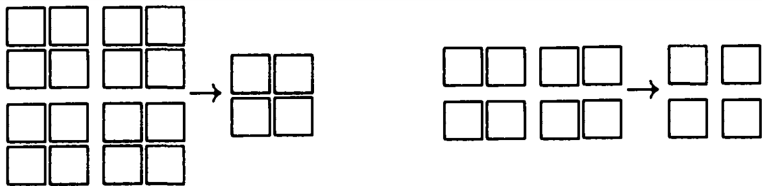
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтинках сірого не проходять цей крок.



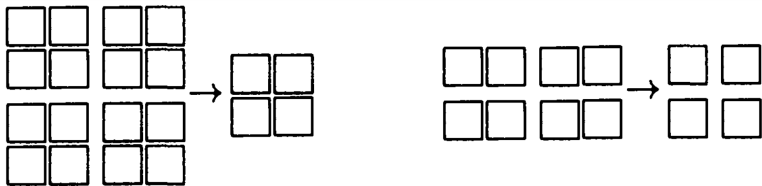
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



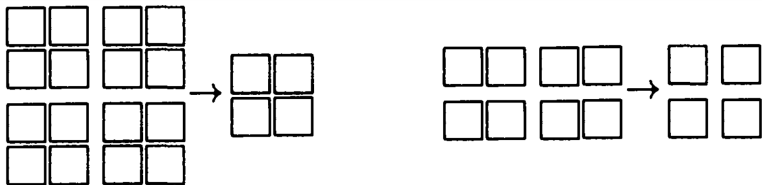
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або “4 : 1 : 1”) або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка “4 : 2 : 2”). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



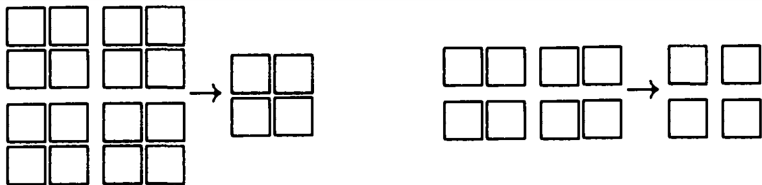
2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.



2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або “4 : 1 : 1”) або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка “4 : 2 : 2”). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтинках сірого не проходять цей крок.



2. Кольорові зображення зменшуються шляхом створення пікселів із низькою роздільною здатністю з вихідних (цей крок використовується лише тоді, коли потрібне ієрархічне стиснення; він завжди пропускається для зображень у градаціях сірого). Зменшення дискретизації не виконується для компоненти яскравості. Зменшення дискретизації (див. рис.) виконується або у співвідношенні 2 : 1 як по горизонталі, так і по вертикалі (так звана дискретизація 2h2v або "4 : 1 : 1") або у співвідношенні 2 : 1 по горизонталі та 1 : 1 по вертикалі (2h1v, або вибірка "4 : 2 : 2"). Оскільки це робиться на двох із трьох компонентів кольору, то 2h2v зменшує зображення до $1/3 + (2/3) \cdot (1/4) = 1/2$ його початкового розміру, тоді як 2h1v зменшує його до $1/3 + (2/3) \cdot (1/2) = 2/3$ початкового розміру. Оскільки компонент яскравості не торкається, то немає помітної втрати якості зображення. Зображення у відтінках сірого не проходять цей крок.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються блоками даних. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній стовпець дублюються останнім рядком і стовпцем. У рядках без нагущення кодів обробляються блоки даних першого компонента зображення, потім блоки даних другого компонента червоної, блок даних третього компонента. У рядках нагущення кодів обробляються три блоки даних (R, G, B) кожного компонента зображення, потім три блоки даних (R, G, B) кожного з них.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок частотних компонентів, вони представляють паралельні значення простору частот) кожної частоти в блоку. Це потім дані зображення в дозволуваного ступеня стиснення інформації. Особливо це стосується трансформантного функції косинус, тоді дані обробляються ступеню стиснення через обчислення точності кожної частоти зростає. Це означає, що простір без останнього ряду 8×8 блоку (або 8×8 блоку даної частоти зображення) зростає, коли дані зберігаються.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

3. Пікселі кожного компонента кольору організовані в групи розміром 8×8 пікселів, які називаються *одиницями даних*. Якщо кількість рядків або стовпців зображення не кратна 8, то нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі блоки даних першого компонента зображення, потім блоки даних другого компонента і, нарешті, блоки даних третього компонента. У режимі чергування кодер обробляє три верхні ліві (#1) блоки даних трьох компонентів зображення, потім три блоки даних (#2) справа від них і т.д.
4. Дискретне косинусне перетворення (DCT) потім застосовується до кожного блоку даних, щоб створити блок даних 8×8 (або блок) частотних компонентів. Вони представляють середнє значення пікселя та послідовні високочастотні зміни в групі. Це готує дані зображення до вирішального етапу втрати інформації. Оскільки DCT включає трансцендентну функцію косинус, то це має спричинити певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) буде деяка втрата якості зображення, але зазвичай вона дуже незначна.

Стиснення зображень. JPEG

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

5. Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке становить частотний коефіцієнт.

6. Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодується за допомогою комбінації 0-255.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яксвітлості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

5. Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування (QC)*, а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG i , в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яксравності та кольоровості (див. табл.).
6. Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

Стиснення зображень. JPEG

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його **коефіцієнтом квантування (QC)**, а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

5. Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яксравності та кольоровості (див. табл.).
6. Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG i , в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG i , в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG i , в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яксравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

- Кожна з 64 частотних складових в одиниці даних ділиться на окреме число, яке називається його *коефіцієнтом квантування* (QC), а потім округлюється до цілого числа. Тут інформація втрачається безповоротно. Великі QC викликають більше втрат, а тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG i , в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці QC, рекомендовані стандартом JPEG для компонентів зображення яскравості та кольоровості (див. табл.).
- Усі 64 квантованих частотних коефіцієнта (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації RLE і кодування Гаффмана. Варіант арифметичного кодування, відомий як кодер QM, можна використовувати замість кодування Гаффмана.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стиснені дані зберігаються в файлі.

8. Третій формат:

Другий формат має сенс у випадках, коли потрібно стиснути зображення та відкодувати дані з однієї таблиці. Третій формат використовує дві таблиці.

9. Четвертий формат: формат для даних стисненого зображення.

10. П'ятий формат: формат для даних стисненої таблиці.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

7. Останній крок додає заголовки та значення всіх використаних параметрів JPEG і виводить результат. Стислий файл може мати один із трьох форматів:
- (1) формат обміну, у якому файл містить стиснене зображення та всі таблиці, необхідні декодеру (переважно таблиці квантування та таблиці кодів Гаффмана),
 - (2) скорочений файл формат для даних стисненого зображення, де файл містить стиснене зображення і може не містити таблиць (або лише кілька таблиць), і
 - (3) скорочений формат для даних специфікації таблиці, де файл містить лише таблиці, а не стиснене зображення.

Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто тим самим кодером, використовуючи однакові таблиці. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, віддаленого від одного файлу з даними специфікації таблиці.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “сканами” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

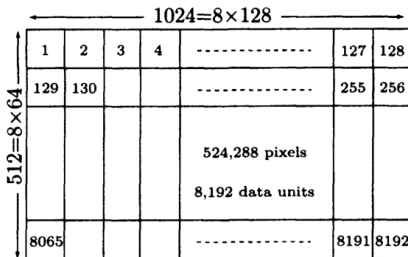
Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

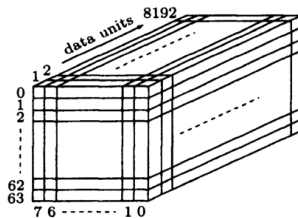
Декодер JPEG робить обернені дії. Отже, JPEG є симетричним методом стиснення.

Прогресуючий режим є опційним для JPEG. У цьому режимі високочастотні коефіцієнти DCT записуються в стислий файл блоками, які називаються “*сканами*” (*scan*). Кожен прочитаний декодером скан дає можливість підправити та уточнити рисунок. Ідея полягає в тому, що перші скани використовуються для швидкого представлення зображення низької якості. Далі відбувається або декодування наступних сканів, або відмова подальшого декодування. Плата за це полягає в тому, що кодер повинен зберігати в буфері всі коефіцієнти всіх одиниць даних до того, як їх надіслати в скан (оскільки вони посилаються в скан у зворотному порядку, а не в порядку їхньої генерації). Крім того, доводиться робити весь процес декодування для кожного скана, що уповільнює прогресуюче декодування.

Стиснення зображень. JPEG



(a)

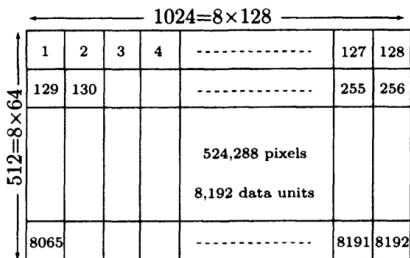


(b)

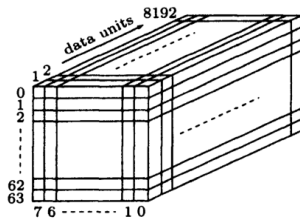
На рис. (а) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

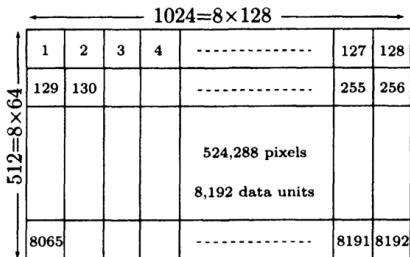


(b)

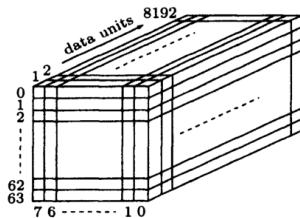
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

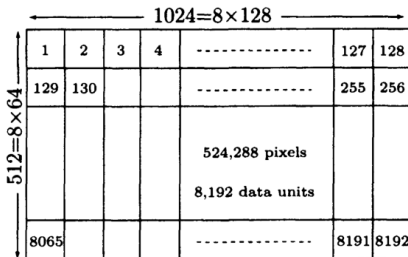


(b)

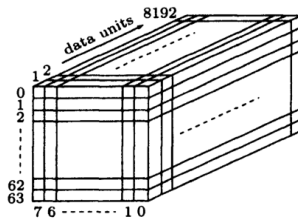
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

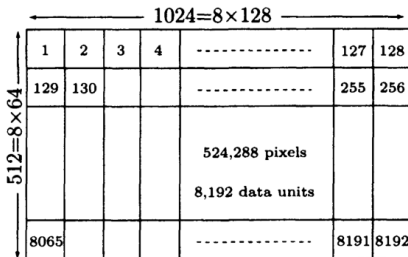


(b)

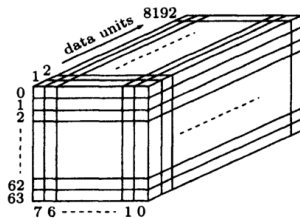
На рис. (а) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

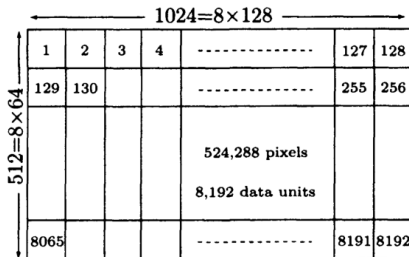


(b)

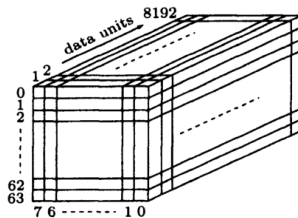
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

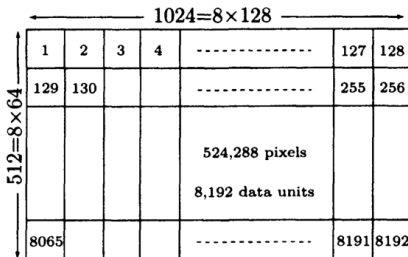


(b)

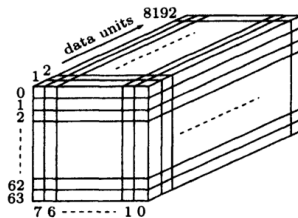
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

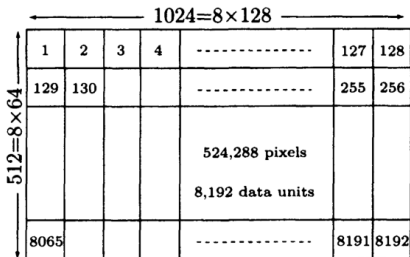


(b)

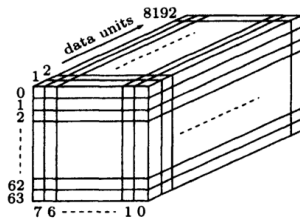
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

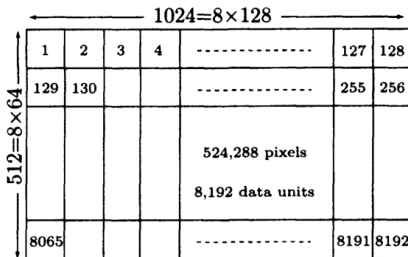


(b)

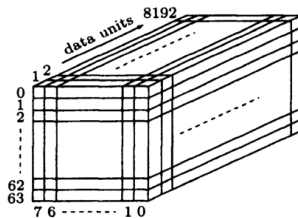
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

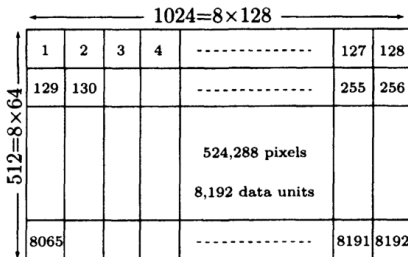


(b)

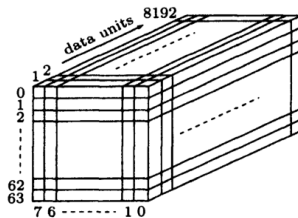
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер запише їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

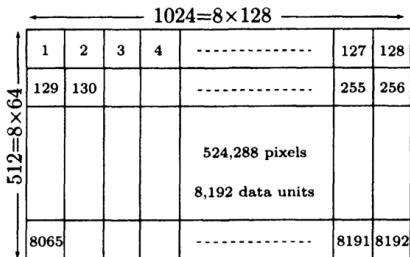


(b)

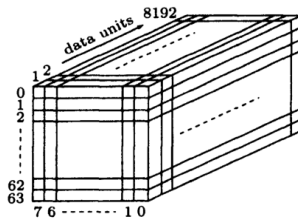
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

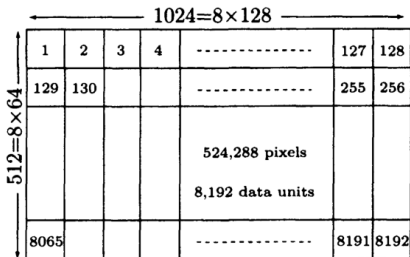


(b)

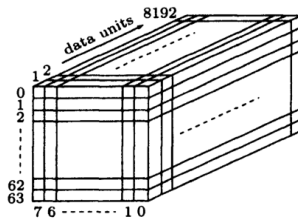
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

Стиснення зображень. JPEG



(a)

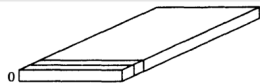


(b)

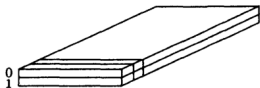
На рис. (a) наведено приклад зображення з роздільною здатністю 1024×512 . Це зображення поділено на $128 \times 64 = 8192$ одиниць даних, кожна з яких перетворена за допомогою DCT в блок із 64 чисел по 8 біт. На рис (b) зображено паралелепіпед, довжина якого дорівнює 8192 одиницям даних, висота дорівнює 64 коефіцієнтам DCT (коефіцієнт DC розташований нагорі з номером 0), а ширина дорівнює 8 біт кожного коефіцієнта.

Після підготовки всіх одиниць даних у буфері пам'яті кодер записує їх у стиснутий файл одним із двох методів: *спектральним вибором* або *послідовним наближенням* (див. рис. (c),(d)).

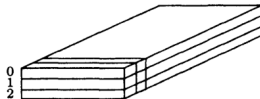
Стиснення зображень. JPEG



1st scan

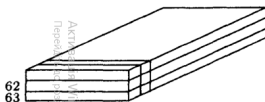


2nd scan



3rd scan

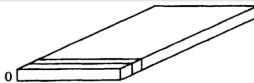
⋮



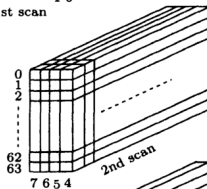
kth scan

Активна зона
Передній план
Настір

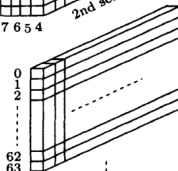
(c)



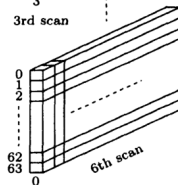
1st scan



2nd scan

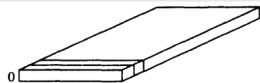


3rd scan

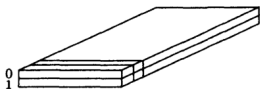


(d)

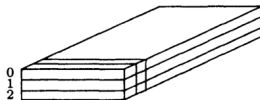
Стиснення зображень. JPEG



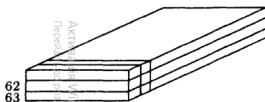
1st scan



2nd scan



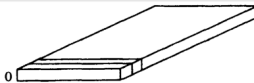
3rd scan



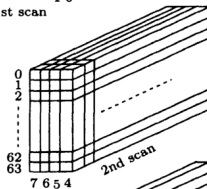
kth scan

АКТУАЛЬНІСТЬ
ПЕРЕВІРИТИ
НАСТРІК

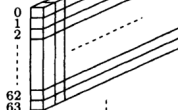
(c)



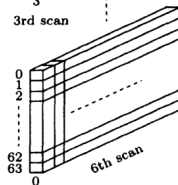
1st scan



2nd scan



3rd scan



(d)

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС.

Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Перше сканування в будь-якому методі — це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, то кожне послідовне сканування складається з кількох послідовних (смуги) коефіцієнтів АС. Якщо використовується послідовне наближення, то друге сканування складається з чотирьох старших бітів усіх коефіцієнтів АС, а кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно).

В ієрархічному режимі кодер декілька разів зберігає зображення у вихідному файлі з кількома роздільними здатностями. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного файлу з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж необхідний для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим стиснення.

Ієрархічний режим корисний у випадках, коли потрібно виводити зображення високої роздільної здатності з низькою роздільною здатністю. Старі матричні принтери можуть бути хорошим прикладом пристрою виводу з низькою роздільною здатністю, який все ще використовується.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Режим JPEG без втрат обчислює “прогнозоване” значення для кожного пікселя, генерує різницю між пікселем і його прогнозованим значенням для відносного кодування, та кодує різницю за допомогою того самого методу (тобто кодування Гаффмана або арифметичне кодування), як у кроці 5 вище. Передбачене значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних лекціях ці кроки розглядатимуться більш детально.

Дякую за увагу!