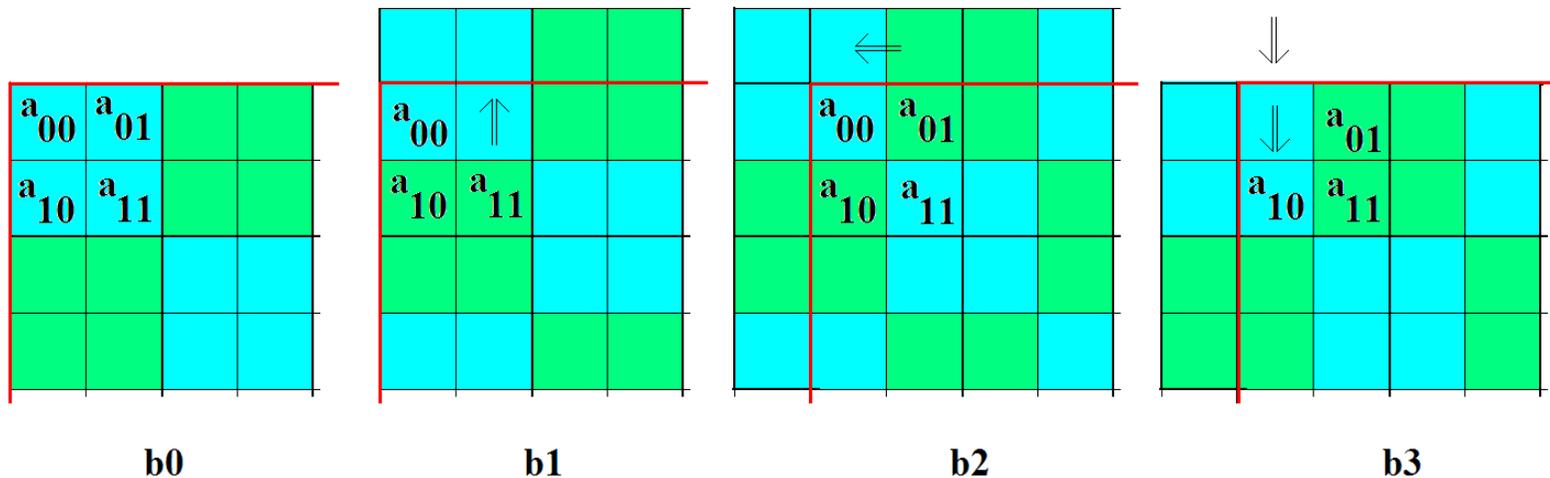


СУБПИКСЕЛЬНОЕ СКАНИРОВАНИЕ И СУБПИКСЕЛЬНАЯ ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

Блажевич С.В., Винтаев В.Н.,
Ушакова Н.Н., Шашков В.В.

*Белгородский государственный университет,
г. Белгород, Россия*



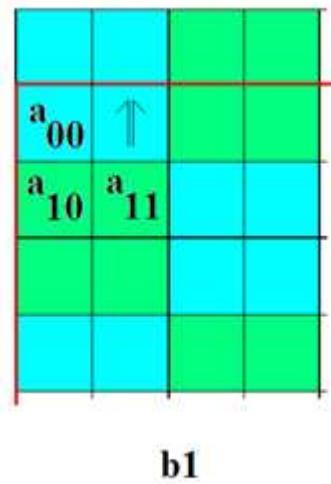
- Рис.1. Схема формирования изображений на крупных пикселях.
 Стрелка указывает направление, в котором происходит сдвиг соответствующей матрицы b_i относительно искомой матрицы a .
 Красными линиями отмечено положение изображения, над которым производится операция оцифровки (соответствующее положению матрицы a и несдвинутой матрицы b_0).

a_{00}	a_{01}		
a_{10}	a_{11}		

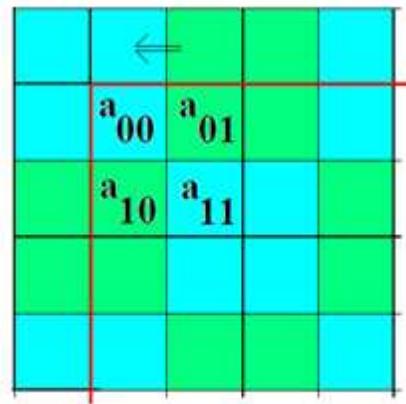
a_{00}	a_{01}		
a_{10}	a_{11}		

b_0

- $b_0(0,0) = a(0,0) + a(0,1) + a(1,0) + a(1,1)$
- $b_0(0,1) = a(0,2) + a(0,3) + a(1,2) + a(1,3)$
- - - - - -

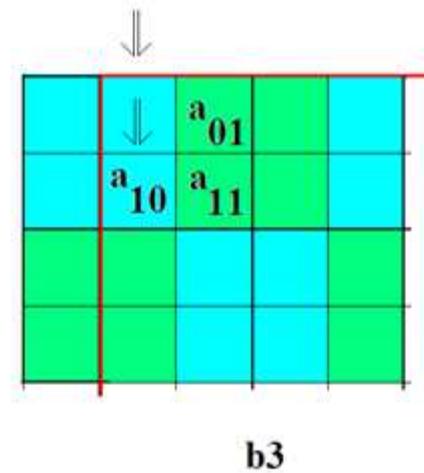


- $b1(0,0)=0 + 0 + a(0,0)+a(0,1)$
- $b1(0,1)=0 + 0 + a(0,2)+a(0,3)$
- - - - - -



b2

- $b2(0,0)=a(0,0)$
- $b2(0,1)=a(0,1)+a(0,2)$
- - - - - -



- $b3(0,0)=a(0,0)+a(1,0)$
- $b3(0,1)=a(0,1)+a(0,2)+a(1,1)+a(1,2)$
- - - - - -

```

AXX(b0, b1, b2, b3) :=
  n ← rows(b3) - 0
  m ← cols(b3)
  a0,0 ← b2,0,0
  a0,1 ← b1,0,0 - a0,0
  a1,0 ← b3,0,0 - a0,0
  a1,1 ← b0,0,0 - (a0,0 + a0,1 + a1,0)
  for j ∈ 1..n - 1
    a2,j,0 ← b2,1,j,0 - a2,j-1,0
    a2,j,1 ← b1,1,j,0 - a2,j,0 - a2,j-1,0 - a2,j-1,1
    a2,j+1,0 ← b3,1,j,0 - a2,j,0
    a2,j+1,1 ← b0,1,j,0 - (a2,j,0 + a2,j+1,0 + a2,j,1)
  for j ∈ 1..m - 1
    a0,2,j ← b2,0,1,j - a0,2,j-1
    a0,2,j+1 ← b1,0,1,j - a0,2,j
    a1,2,j ← b3,0,1,j - a0,2,j - a0,2,j-1 - a1,2,j-1
    a1,2,j+1 ← b0,0,1,j - (a0,2,j + a0,2,j+1 + a1,2,j)
  for i ∈ 1..n - 1
    for j ∈ 1..m - 1
      a2,i,2,j ← b2,i,j - (a2,i-1,2,j + a2,i,2,j-1 + a2,i-1,2,j-1)
      a2,i,2,j+1 ← b1,i,j - (a2,i,2,j + a2,i-1,2,j + a2,i-1,2,j+1)
      a2,i+1,2,j ← b3,i,j - (a2,i,2,j + a2,i,2,j-1 + a2,i+1,2,j-1)
      a2,i+1,2,j+1 ← b0,i,j - (a2,i,2,j + a2,i,2,j+1 + a2,i+1,2,j)
  a

```

- **Рис.55. Алгоритм обратной задачи (восстановление изображения на малых пикселях)**

В качестве демонстрации
работоспособности алгоритма решения
обратной задачи, реализованного
программно в рамках пакета MathCad,
рассмотрим растровое изображение
высокого разрешения, полученное с
помощью спутниковой системы Quik
Bird



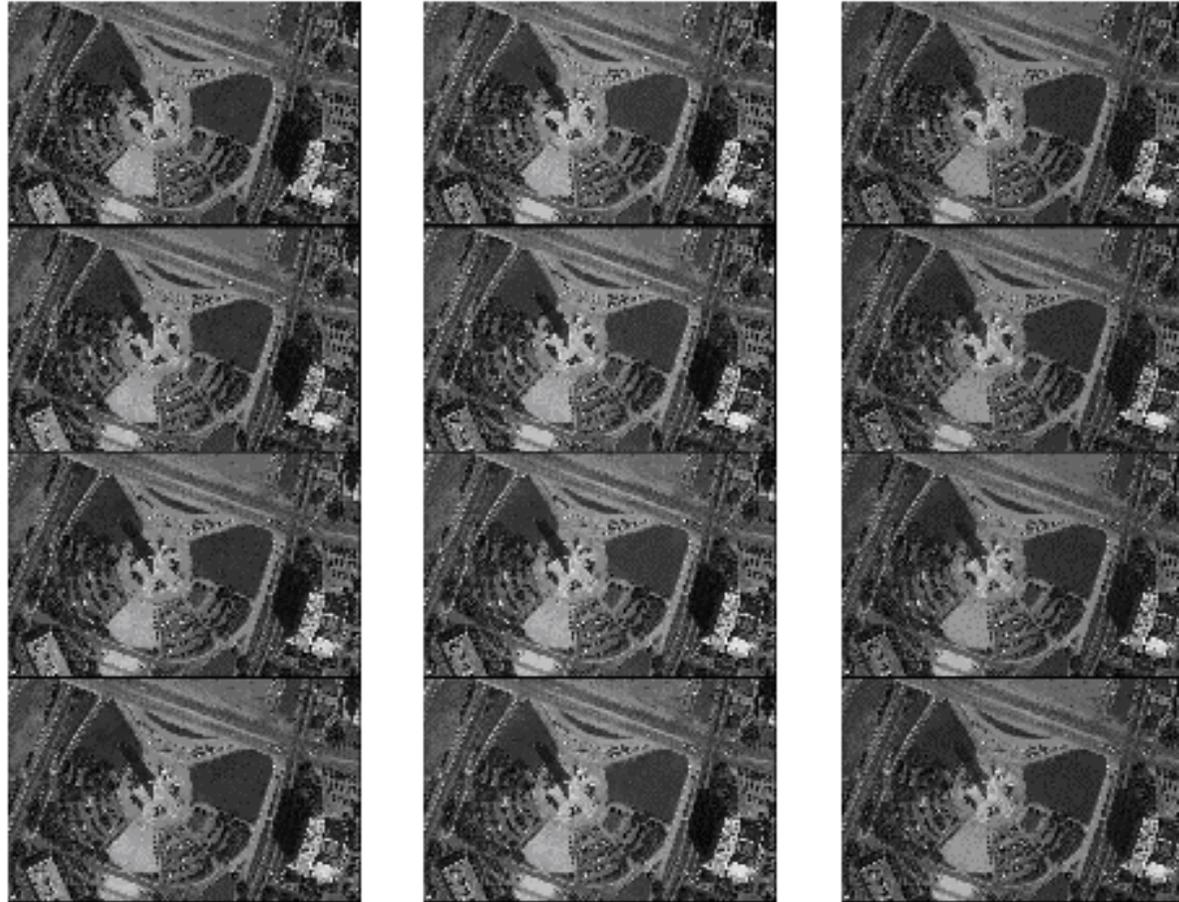
- Рис.2. Снимок ландшафта со спутника Quik Bird (город Гавана).

```

XXX(a) :=
  n ← rows(a) / 2
  m ← cols(a) / 2
  b00,0 ← a0,0 + a0,1 + a1,0 + a1,1
  b10,0 ← a0,0 + a0,1
  b20,0 ← a0,0
  b30,0 ← a0,0 + a1,0
  for j ∈ 1..n - 1
    b21,j,0 ← a2-j-1,0 + a2-j,0
    b11,j,0 ← a2-j,0 + a2-j-1,0 + a2-j-1,1 + a2-j,1
    b31,j,0 ← a2-j,0 + a2-j+1,0
    b01,j,0 ← a2-j,0 + a2-j+1,0 + a2-j,1 + a2-j+1,1
  for j ∈ 1..m - 1
    b20,1,j ← a0,2-j + a0,2-j-1
    b10,1,j ← a0,2-j + a0,2-j+1
    b30,1,j ← a0,2-j + a0,2-j-1 + a1,2-j-1 + a1,2-j
    b00,1,j ← a0,2-j + a0,2-j+1 + a1,2-j + a1,2-j+1
  for i ∈ 1..n - 1
    for j ∈ 1..m - 1
      b0i,j ← a2-i,2-j + a2-i,2-j+1 + a2-i+1,2-j + a2-i+1,2-j+1
      b1i,j ← a2-i,2-j + a2-i,2-j+1 + a2-i-1,2-j + a2-i-1,2-j+1
      b2i,j ← a2-i,2-j + a2-i-1,2-j + a2-i,2-j-1 + a2-i-1,2-j-1
      b3i,j ← a2-i,2-j + a2-i+1,2-j + a2-i,2-j-1 + a2-i+1,2-j-1
  b ← stack[(b0),(b1),(b2),(b3)]

```

- **Рис.1 Программа создание 4-х сдвинутых изображений на крупных пикселях (прямая задача преобразования)**



- **Рис. 3. четыре изображения (для каждого цвета) со субпиксельным сдвигом (на пол пикселя), сформированные путем решения указанной выше прямой задачи.**



- **Рис.5. Восстановленное изображение**



- **Рис.5 Сравнение фрагментов изображений на больших и малых (восстановленного) пикселях**

Спасибо за внимание