Γεωμετρικοί μετασχηματιμοί εικόνας

Μάθημα: Υπολογιστική Οραση

Κ. Δελήμπασης

Γεωμετρικοί Μετασχηματισμοί

- Ορισμός σημείου στον Ευκλείδιο χώρο: $\mathbf{p}=[x_p,y_p,z_p]^T$, όπου x_p, y_p, z_p πραγματικοί αριθμοί.
- Εστω E^3 το σύνολο των **p**.
- Ενας γεωμετρικός μετασχηματισμός Τ(**π**), με διάνυσμα παραμέτρων π, ορίζεται ως: Τ: E^3 --> E^3
- Οι μετασχηματισμοί σε 2D αποτελούν υποπερίπτωση των 3D.
- Παραδείγματα: Μεταφορά (translation), περιστροφή (rotation), αλλαγή κλίμακας (scaling).
- Κάθε γεωμετρικός μετασχηματισμός που μπορεί να περιγραφεί σαν συνδυασμός μεταφορών, περιστροφώ, ή αλλαγής κλίμακας λέγεται γενικευμένος συσχετισμένος μετασχηματισμός (affine).

Ομογενείς συντεταγμένες

- Σύνθεση μετασχηματισμών
  - Ακολουθία μετασχηματισμών εκφράζεται σαν γινόμενο των αντίστοιχων πινάκων, με τον πίνακα του 1ου μετασχηματισμού αριστερά
  - Η μετατόπιση είναι ο μόνος μετασχηματισμός ο οποίος απαιτεί πρόσθεση
- Ομογενείς συντεταγμένες: για κάθε σημείο P(x,y), εισάγουμε μία επιπλέον συντεταγμένη w, $P(x,y,w_0)$,
  - Το σημείο $P'(x/w_0, y/w_0, 1)$ αποτελεί την αναπαράσταση ομογενών συντεταγμένων στο επίπεδο $w=w_0$
  - Συνήθως χρησιμοποιείται η βασική αναπαράσταση με $w_0=1$
- Με την εφαρμογή των ομογενών συντεταγμένων η μετατόπιση γίνεται με πολλαπλασιασμό πινάκων και έτσι είναι δυνατή η σύνθεση πολλών διαδοχικών μετασχηματισμών σε ένα μόνο πίνακα μετασχηματισμού.

- Συχνά είναι απαραίτητο να εφαρμόσουμε ένα συσχετισμένο μετασχηματισμό σε μία εικόνα, πχ
  – Περιστροφή της εικόνας γύρω από το κέντρο μάζας της
  – Αλλαγή κλίμακας της εικόνας (ισοδύναμα interpolation).
- Εστω *T* ο μετασχηματισμός T: E^3 --> E^3 εικόνας που μετρέπει μία εικόνα *I* στην εικόνα I_1.

$$I_1\left(T\left(x, y\right)\right) = I\left(x, y\right)$$

$$T\left(x, y\right) = \mathbf{T}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **T** είναι ο πίνακας του μετασχηματισμού (σε ομογενείς συντεταγμένες), ο οποίος εφαρμοζόμενος στις συντεταγμένες ενός pixel παράγει τις συντεταγμένες του αντίστοιχου pixel της μετασχηματισμένης εικόνας.

- Μετατώπιση (Translation) κατά d_x, d_y:

$$T = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$$

- Αλλαγή κλίμακας (ανεξάρτητα σε κάθε διάσταση), ως προς την αρχή των αξόνων.

$$T(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Περιστροφή (Rotation) γύρω από τον άξονα των *z*, ως προς την αρχή των αξόνων.

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Οποιοσδήποτε συσχετισμένος μετασχηματισμός (στις 2 διαστάσεις) μπορεί να περιγραφεί από ένα πίνακα της μορφής:

$$\mathbf{T} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

- Ο μετασχηματισμός Affine εφαρμόζεται ως εξής:
  - Για κάθε pixel (x1,y1) της I1, υπολογίζονται οι νέες συντεταγμένες (x2,y2) ως εξής:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- Επισημαίνεται ότι:
  - Τα (x2,y2) δεν είναι πάντα ακέραια, μπορεί να πάρουν και αρνητικές τιμές, ή τιμές μεγαλύτερες από τον αριθμό γραμμών και στηλών της I1.
  - Εφαρμογή του μετασχηματισμού σε Matlab:
  - `I2=imtransform(I1,T,'nearest','XData',XData,'YData',YData,'FillValues',0);`

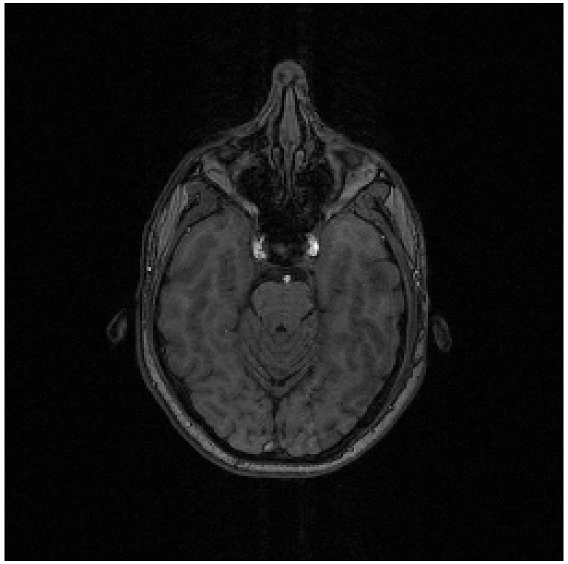
Παράδειγμα

- Εστω ότι επιθυμούμε να μετατοπίσουμε μία εικόνα μεγέθους 256x256 κατά 10,-8 στους δύο άξονες, να την περιστρέψουμε κατά $\vartheta=6^0$ και να αλλάξουμε την κλίμακα στον άξονα *X* και *Y* κατά 1.2 και 0.8 αντίστοιχα.

$$T = \underbrace{\begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{(0,0)−−>κέντρο μάζας}}} \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας αλλαγής} \\ \text{κλίμακας}}} \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Πίνακας περιστροφής}} \underbrace{\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{dx, dy}}} \underbrace{\begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{κέντρου μάζας --> (0,0)}}}$$

$$T = \underbrace{\begin{bmatrix} 1 & 0 & 128 \\ 0 & 1 & 128 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{(0,0)−−>κέντρο μάζας}}} \underbrace{\begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας αλλαγής} \\ \text{κλίμακας}}} \underbrace{\begin{bmatrix} \cos\left(\frac{6\pi}{180}\right) & -\sin\left(\frac{6\pi}{180}\right) & 0 \\ \sin\left(\frac{6\pi}{180}\right) & \cos\left(\frac{6\pi}{180}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Πίνακας περιστροφής +6 μοιρών}} \underbrace{\begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & -8 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{dx, dy}}} \underbrace{\begin{bmatrix} 1 & 0 & -128 \\ 0 & 1 & -128 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Πίνακας μετατόπισης} \\ \text{κέντρου μάζας --> (0,0)}}}$$

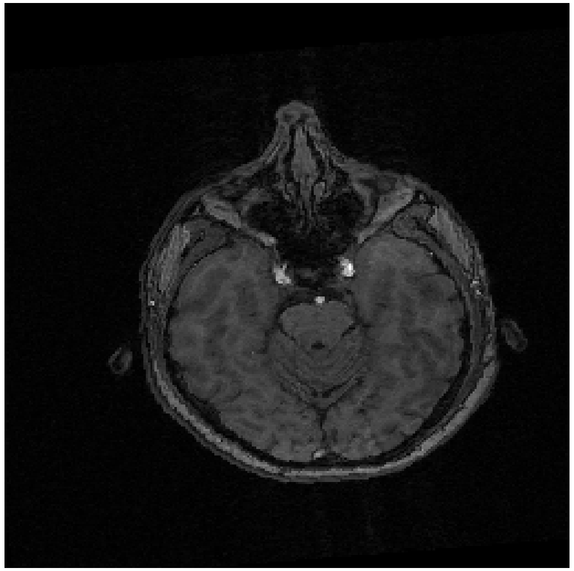
- Εφαρμογή του μετασχηματισμού του προηγούμενου παραδείγματος



Αρχική εικόνα

$$T = \begin{bmatrix} 1.1934 & -0.1254 & -8.7030 \\ 0.0836 & 0.7956 & 15.4572 \\ 0 & 0 & 1 \end{bmatrix}$$

Πίνακας Μετασχηματισμού



Μετασχηματισμένη εικόνα

Απλός αλγόριθμος εφαρμογής συσχετισμένου μετασχηματισμού: μη ενδεικνυόμενος

Εστω ότι θέλουμε να μετασχηματίσουμε γεωμετρικά δοθείσα εικόνα I_1, βάσει συσχετισμένου μετασχηματισμού με πίνακα A.

Βήμα 1ο: Προσδιορίζουμε τη προβολή (απεικόνιση) του κέντρου του κάθε pixel (x,y) της αρχικής εικόνας I_1 στη νέα εικόνα I_2 : $(x_1,y_1,1)^T = A\cdot(x,y,1)^T$.

Βήμα 2ο: Βρίσκουμε το pixel της νέας εικόνας I_2 του οποίου το κέντρο βρίσκεται πιο κοντά στο σημείο της προβολής (στρογγυλοποίηση συντεταγμένων)

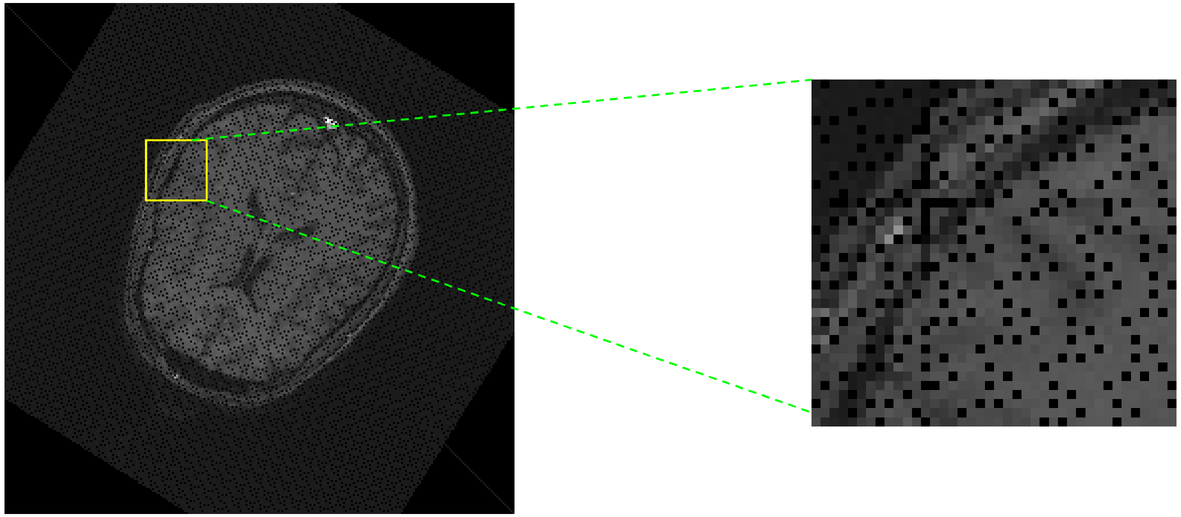
Βήμα 3ο: Θέτουμε τη φωτεινότητα του ζητούμενου pixel της τελικής εικόνας ίση με τη φωτεινότητα του pixel, από το βήμα 2, της αρχικής εικόνας:

$I_2(\mathrm{round}(x_1),\mathrm{round}(y_1)) = I_1(x,y)$.

Παράδειγμα ψευδοκώδικα του προηγούμενου αλγόριθμου για περιστροφή εικόνας κατά γωνία θ γύρω από το κέντρο μάζας της CM.

```
A=T(-CM)R(θ)T(CM)
for i=1:256
   for j=1:256
        [i1,j1]^T=A*[i,j]^T
        if i1>256 →i1=256;
        if i1<=1→i1=1;
        if j1>256 →j1=256;
        if j1<=1→j1=1;
        IM(round(i1),round(j1))=I(i,j);
   end;
end;
```

Παράδειγμα περιστροφής εικόνας με χρήση του προηγούμενου αλγόριθμου



Παρατηρούμε το artifact των μηδενικών Pixel (τα οποία δεν έχουν πάρει τιμές). Η χρήση της παρεμβολής βάσει του κοντινότερου γείτονα είναι πολύ απλή στην υλοποίηση αλλά έχει το μειονέκτημα ότι θολώνει την περιεστραμένη εικόνα. Εναλλακτικά μπορεί να χρησιμοποιηθεί η διγραμμική παρεμβολή.

Γενικός αλγόριθμος εφαρμογής συσχετισμένου μετασχηματισμού

- *Βήμα 1ο*: Προσδιορίζουμε τη προβολή (απεικόνιση) του κέντρου του κάθε pixel (x,y) της νέας εικόνας I2 στην αρχική εικόνα I_1 :

  $(x_1,y_1,1)^T=A^{-1}\cdot(x,y,1)^T.$

Ύπολογίζουμε την τιμή του (x,y) της νέας εικόνας I_2 με 2 τρόπους:

- *Βήμα 2α*: **Παρεμβολή κοντινότερου γείτονα**: Βρίσκουμε το pixel της αρχικής εικόνας του οποίου το κέντρο βρίσκεται πιο κοντά στο σημείο της προβολής με στρογγυλοποίηση:

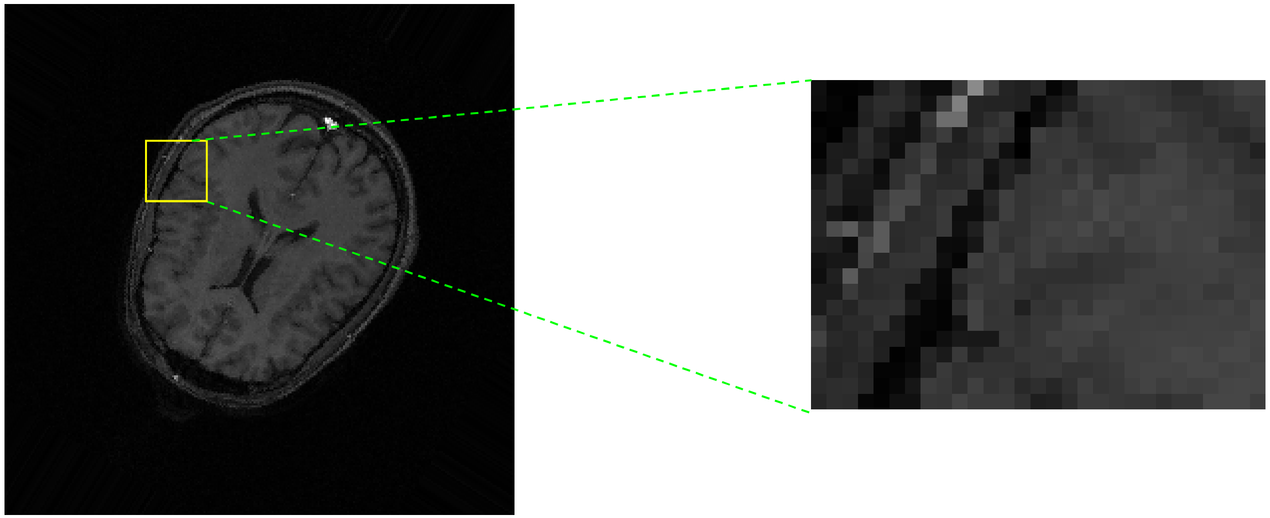
  $I_2(x,y)$ = I_1(round(x_1),round(y_1))
- *Βήμα 2β*: **Διγραμμική Παρεμβολή**: Θέτουμε την τιμή του (x_1,y_1) ζητούμενου pixel της τελικής εικόνας I_2, ίση με την τιμή της αρχικής εικόνας I_1 στη θέση (x,y). Επειδή τα (x_1,y_1) είναι εν γένει πραγματικοί αριθμοί, εφαρμόζουμε διγραμμική παρεμβολή:

$$I_2(y,x)=I_1(y_1,x_1)=I_1([y_1]+a,[y_1]+b)=$$
$$(1-a)(1-b)I([y_1],[x_1])+a(1-b)I([y_1],[x_1]+1)+(1-a)bI_1([y_1]+1,[x_1])$$
$$+abI_1([y_1]+1,[x_1]+1)$$

Αλγόριθμος περιστροφής εικόνας για διόρθωση artifact, βάσει του κοντινότερου γείτονα

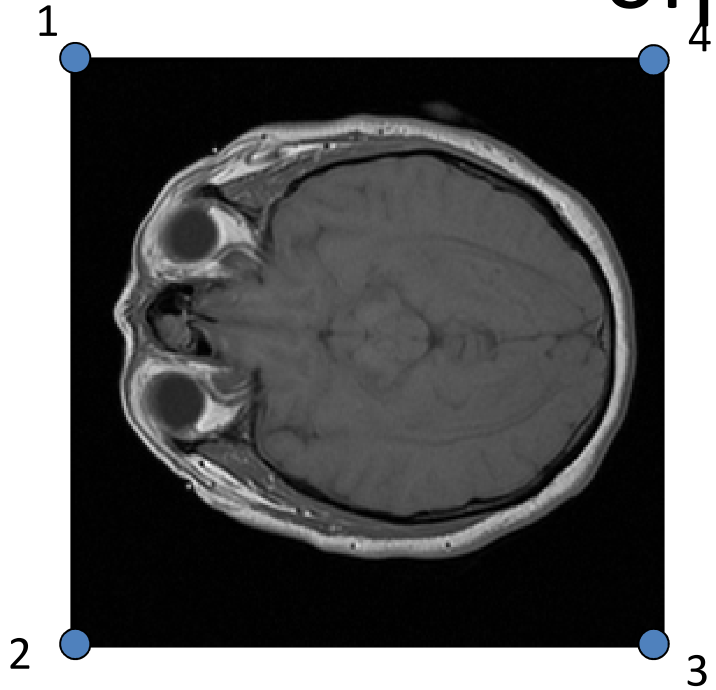
```
A=T(-CM)R(θ)T(CM)              // Υπολογισμός του πίνακα περιστροφής A
for i=1:256                    // Για κάθε pixel της νέας εικόνας I2
   for j=1:256
        [i1,j1]^T=(A^-1)*[i,j]^T    // Υπολογισμός του pixel της I1 από το οποίο
                                    // προέρχεται με χρήση του αντίστροφου του A
        if i1>256 →i1=256;
        if i1<=1→i1=1;              // Ελεγχος αν το (i1,j1) είναι εντός
        if j1>256 →j1=256;          // της αρχικής εικόνας
        if j1<=1→j1=1;
        IM2(i,j)=
      IM(round(i1),round(j1));      // Μέθοδος του κοντινότερου γείτονα
   end;
end;
```

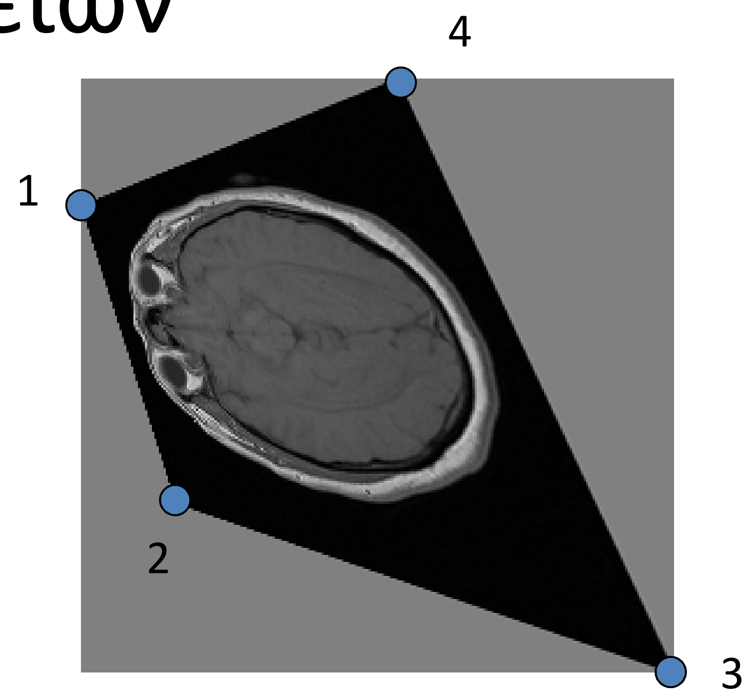
Παράδειγμα περιστροφής εικόνας χωρίς artifacts με χρήση του προηγούμενου αλγόριθμου



Παρατηρούμε ότι το artifact των μηδενικών Pixel έχει διορθωθεί. Η χρήση της παρεμβολής βάσει του κοντινότερου γείτονα είναι πολύ απλή στην υλοποίηση αλλά έχει το μειονέκτημα ότι θολώνει την περιεστραμένη εικόνα. Εναλλακτικά μπορεί να χρησιμοποιηθεί η διγραμμική παρεμβολή.

Παραμόρφωση εικόνας με χρήση 4 σημείων





- Μετασχηματισμοί όπως διγραμμικός και προβολικός απαιτούν τουλάχιστον 4 ζεύγη ομόλογων σημείων

- Γενική μορφή μετασχηματισμού
$$x_1 = F_1(x, y)$$
$$y_1 = F_2(x, y)$$

- Διγραμμικός (δεν διατηρεί ευθείες)
$$x_1 = a_0 + a_1 x + a_2 y + a_3 xy$$
$$y_1 = b_0 + b_1 x + b_2 y + b_3 xy$$

- Προβολικός (διατηρεί ευθείες)
$$x_1 = \frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}, y_1 = \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y}$$

- Οι παράμετροι του μετασχηματισμού υπολογίζονται από τα 4 σημεία με τις μετατοπίσεις τους

Γεωμετρικοί μετασχηματισμοί ομογενών συντεταγμένων σε 3 διαστάσεις (3D)

- Μετατώπιση,

$$\mathbf{T}(\mathbf{d}) = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Αντίστροφος μετασχηματισμός

$$\mathbf{T}^{-1}(\mathbf{d}) = \mathbf{T}(-\mathbf{d})$$

- Αλλαγή κλίμακας:

$$\mathbf{S} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Αντίστροφος μετασχηματισμός

$$\mathbf{S}^{-1}\left(s_x, s_y, s_z\right) = \mathbf{S}\left(s_x^{-1}, s_y^{-1}, s_z^{-1}\right)$$

Περιστροφή γύρω από τους 3 άξονες: Γωνίες Euler

- Πίνακες περιστροφής γύρω από τους άξονες X, Y, Z κατά γωνία θ. Ομογενείς συντεταγμένες.

- Αντίστροφος μετασχηματισμός $R_x(-\theta_x)$
- Προφανώς $R_x(\theta_x)\, R_x(-\theta_x)$=*I*

$$\mathbf{R_x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R_y} = \begin{pmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_z = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Γενική μορφή affine 3D με ομογενείς συντεταγμένες

Μετασχημ. σημείο

Συντεταγμ. σημείου

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

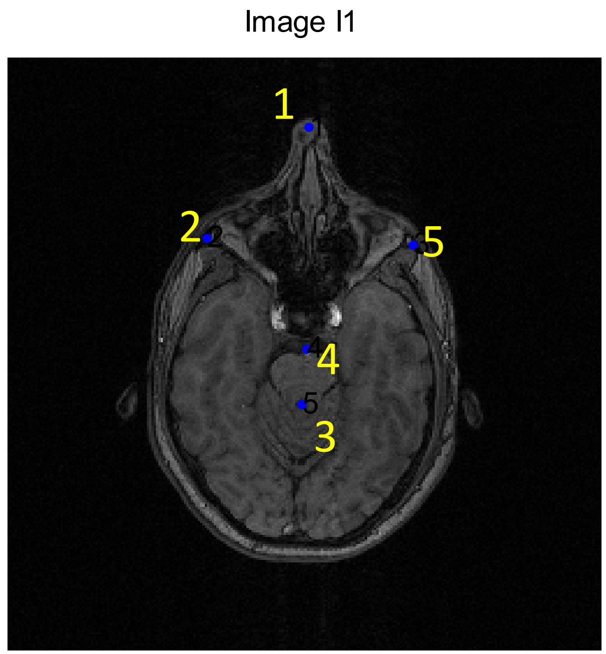
Πίνακας Affine Μετασχ.

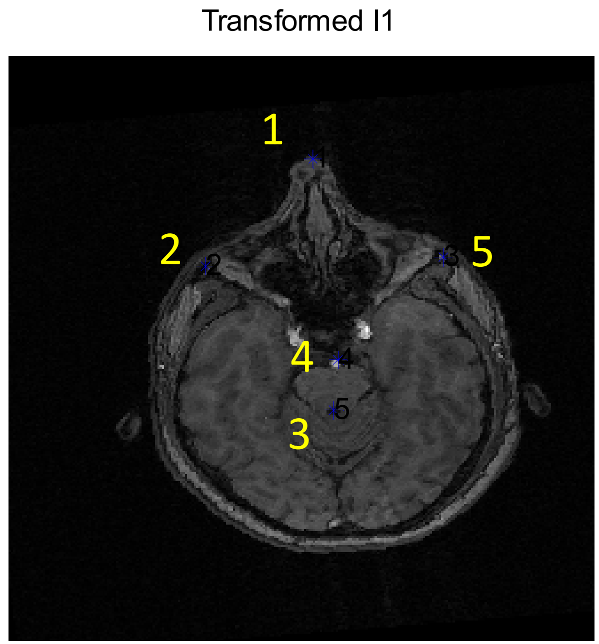
- Για να μετασχηματίσουμε *N* σημεία:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \ldots & x_N \\ y_1 & y_2 & \ldots & y_N \\ z_1 & z_2 & \ldots & z_N \\ 1 & 1 & \ldots & 1 \end{pmatrix}$$

Χωρική ταύτιση εικόνων: Καθορισμός του πίνακα μετασχηματισμού βάσει ομόλογων σημείων

- Εστω 2 εικόνες I1, I2 του ίδιου αντικειμένου, που έχουν συλλεχθεί υπό διαφορετική γεωμετρία.
- Εστω ένας αριθμός από ζεύγη **ομόλογων** σημείων μεταξύ δύο εικόνων: $\{\mathbf{p}_i^A\}$ στην I1 και $\{\mathbf{p}_i^B\}$ στην I2.
- Ζητείται ο πίνακας που μετασχηματίζει γεωμετρικά την I1 στην I2, ώστε τα μετασχηματισμένα σημεία $\{\mathbf{p}_i^A\}$ να συμπίπτουν με τα $\{\mathbf{p}_i^B\}$.
  - Λέμε τότε ότι οι δύο εικόνες ταυτίζονται χωρικά (spatial registration).
- Τα ζεύγη ομόλογων σημείων $\{\mathbf{p}_i^A\}$ στην I1 και $\{\mathbf{p}_i^B\}$ στην I2, $i=1,\ldots,N$ $N>3$, ορίζονται είτε από το χρήστη είτε από κάποια αυτόματη μέθοδο.
- **Ομόλογα** είναι δύο σημεία πάνω στα ίδια αντικείμενα στις δύο διαφορετικές εικόνες.
  - Οι εικόνες δεν ταυτίζονται χωρικά, άρα οι συντεταγμένες δύο ομόλογων σημείων δεν θα είναι ίδιες (πχ η μύτη του ασθενή στην I1 δεν βρίσκεται στα pixel στα οποία βρίσκεται η μύτη του ίδιου ασθενή στην I2).





- Παράδειγμα δύο εικόνων Ι1 και Ι2 του ιδίου αντικειμένου (MRI εγκεφάλου) με 5 ζεύγη ομολόγων σημείων που έχουν τοποθετηθεί σε κοινές ανατομικές δομές από τον χρήστη.

- Για να καθορίσουμε τον μετασχηματισμό Affine χρειαζόμαστε τον πίνακα του μετασχηματισμού ο οποίος έχει 6 αγνώστους:

$$a_{00}, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}$$

- Οι 3 άγνωστοι α_{00}, α_{01}, α_{02} υπολογίζονται από τις X συντεταγμένες των ζευγών ομολόγων σημείων και
- οι 3 άγνωστοι α_{10}, α_{11}, α_{12} υπολογίζονται από τις Y συντεταγμένες.
- Αν N>3 (συνήθης περίπτωση) τότε τα 2 γραμμικά συστήματα είναι υπερκαθορισμένα.

- Εστω οι πίνακες

$$A = \begin{bmatrix} x_1^A & y_1^A & 1 \\ x_2^A & y_2^A & 1 \\ \dots & \dots & \dots \\ x_N^A & y_2^A & 1 \end{bmatrix}$$

$$p = \left(a_{00}, a_{01}, a_{02}\right)^T, q = \left(a_{10}, a_{11}, a_{12}\right)^T$$

$$\mathbf{b}_1 = \left(x_1^A, x_2^A, \dots, x_N^A\right)^T, \mathbf{b}_2 = \left(x_1^B, x_2^B, \dots, x_N^B\right)^T$$

- Πρέπει να επιλυθούν τα γραμμικά συστήματα

$$\mathbf{Ap} = \mathbf{b}_1$$
$$\mathbf{Aq} = \mathbf{b}_2$$

- Ο A είναι διαστάσεων Nx3 ενώ τα **p**,**q** είναι διαστάσεων 3x1 και τα $\mathbf{b}_1$,$\mathbf{b}_2$ είναι δαστάσεων Nx1.
  - Μπορεί να χρησιμοποιηθεί ο τελεστής «\» του Matlab: p=A\b1 και q=A\b2.
- Η παραπάνω λύση ισοδυναμεί με τη μέθοδο ελαχίστων τετραγώνων:

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{p} = \mathbf{A}^T\mathbf{b}_1$$
$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{q} = \mathbf{A}^T\mathbf{b}_2$$

Ελαστικοί γεωμετρικοί μετασχηματισμοί

- Πολλά είδη μετασχηματισμών δε προσεγγίζονται από ολικούς (global) μετασχηματισμούς, όπως affine, bilinear, projective κλπ
- Ελαστική παραμόρφωση λόγω
  - Αναπνοής, κίνησης καρδιάς, μεταβολή σχήματος στομάχου, ουροδόχος κύστη κλπ
- Ο πιο γνωστός ελαστικός γεωμετρικός μετασχηματισμός είναι το μοντέλο TPS (Thin Plate Splines) (Bookstein 1989).

Το μοντέλο TPS (Thin Plate Splines)

- Εστω ότι έχουμε επιλέξει 2 σύνολα ομόλογων σημείων $\{x_i, y_i\}$, $\{x_i', y_i'\}$
- Κατασκευάζουμε τους πίνακες P, Y

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \cdots & \cdots & \cdots \\ 1 & x_n & y_n \end{bmatrix}, 3 \times n; \qquad V = \begin{bmatrix} x_1' & x_2' & \cdots & x_n' \\ y_1' & y_2' & \cdots & y_n' \end{bmatrix}$$

- Ορίζουμε την συνάρτηση $U(r)=r^2 \log r^2 : R^+ \rightarrow R^+$ και κατασκευάζουμε τον πίνακα K, όπου r_{ij} η απόσταση των σημείων i, j.

$$K = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \cdots & \cdots & \cdots & \cdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix}, n \times n;$$

- Κατασκευάζουμε τοn πίνακα L

$$L = \left[\begin{array}{c|c} K & P \\ \hline P^T & O \end{array}\right], (n+3) \times (n+3)$$

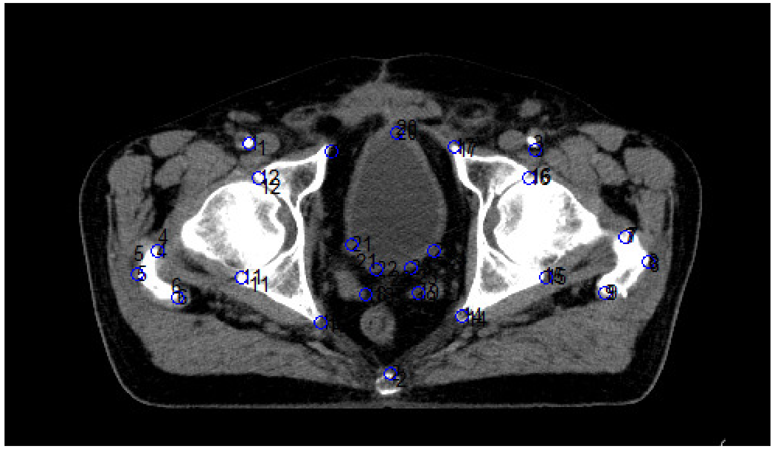
- Κατασκευάζουμε τον πίνακα Y $= (V|0\ 0\ 0)^T$

$$L^{-1}Y = (W|a_1\ \ a_x\ \ a_y)^T$$

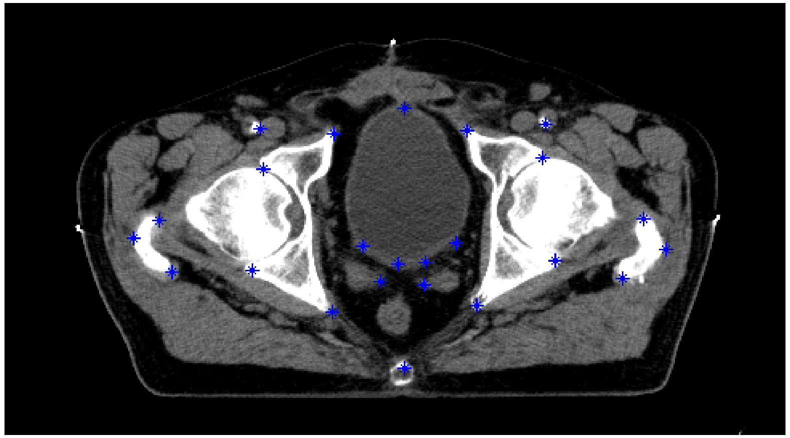
- Υπολογίζουμε για κάθε σημείο (x,y) τις νέες του συντεταγμένες βάσει της ακόλουθης διανυσματικής συνάρτησης:

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i U(|P_i - (x, y)|)$$

Παράδειγμα: ταύτιση εικόνων με ελαστικό μετασχηματιισμό

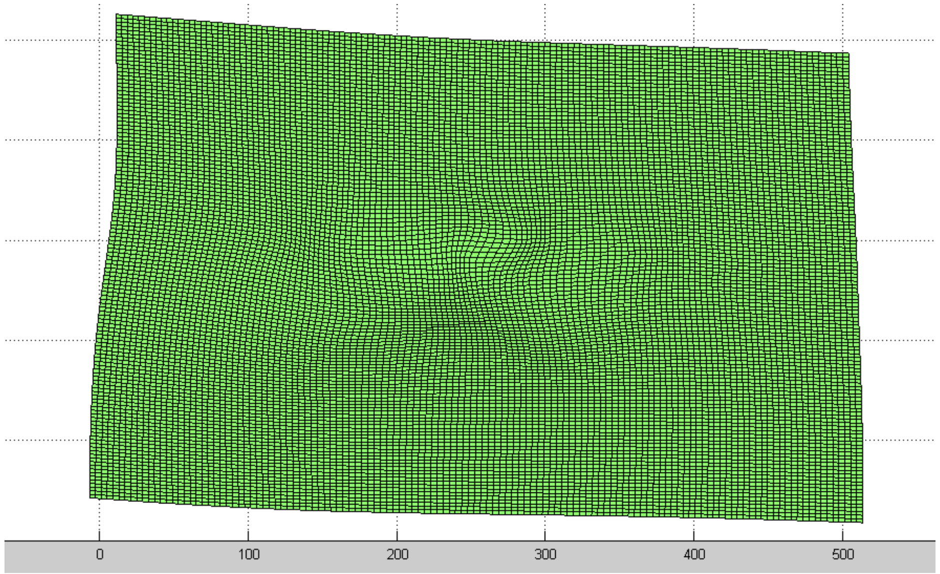


(α)



(β)

- Εστω δύο εικόνες του ίδιου αντικειμένου που έχει παραμορφωθεί ελαστικά.
- Ορίζουμε ζεύγη ομόλογων σημείων στις εικόνες
- Άναζητούμε τον μετασχηματισμό που παραμορφώνει τη (α) ώστε να ταυτιστεί με την (β)



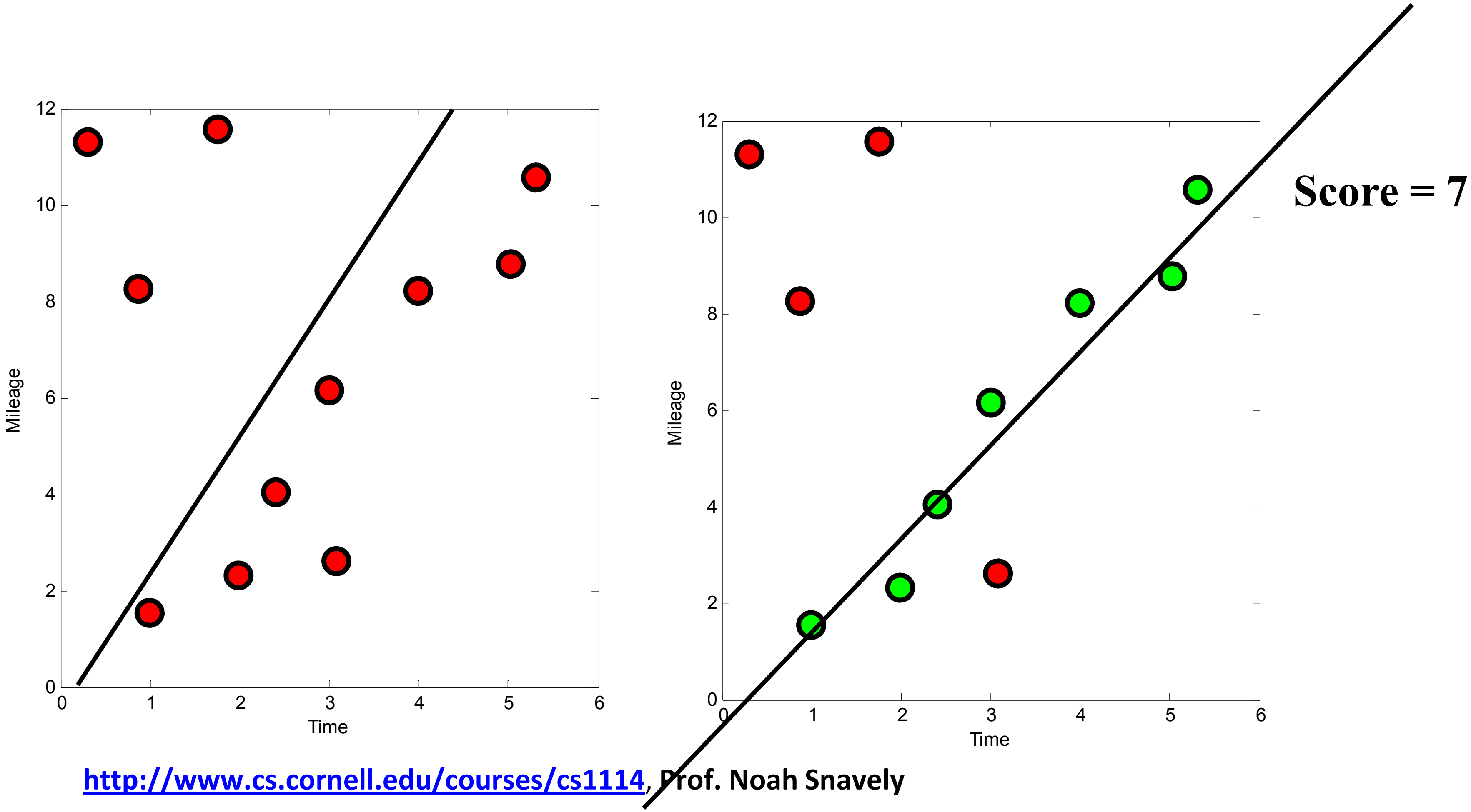
Η εφαρμογή του ελαστικού μετασχηματσμού σε συνθετική εικόνα

Οι ακμές της I2 μετά την εφαρμογή του ελαστικού μετασχηματσμού προβεβλημένες επί της I1

RANSAC

- A more robust approach for parameter estimation of geometric transformation
  - Immune to outliers
  - Can be applied to many geometric models
  - Easy to code
  - Computationally efficient

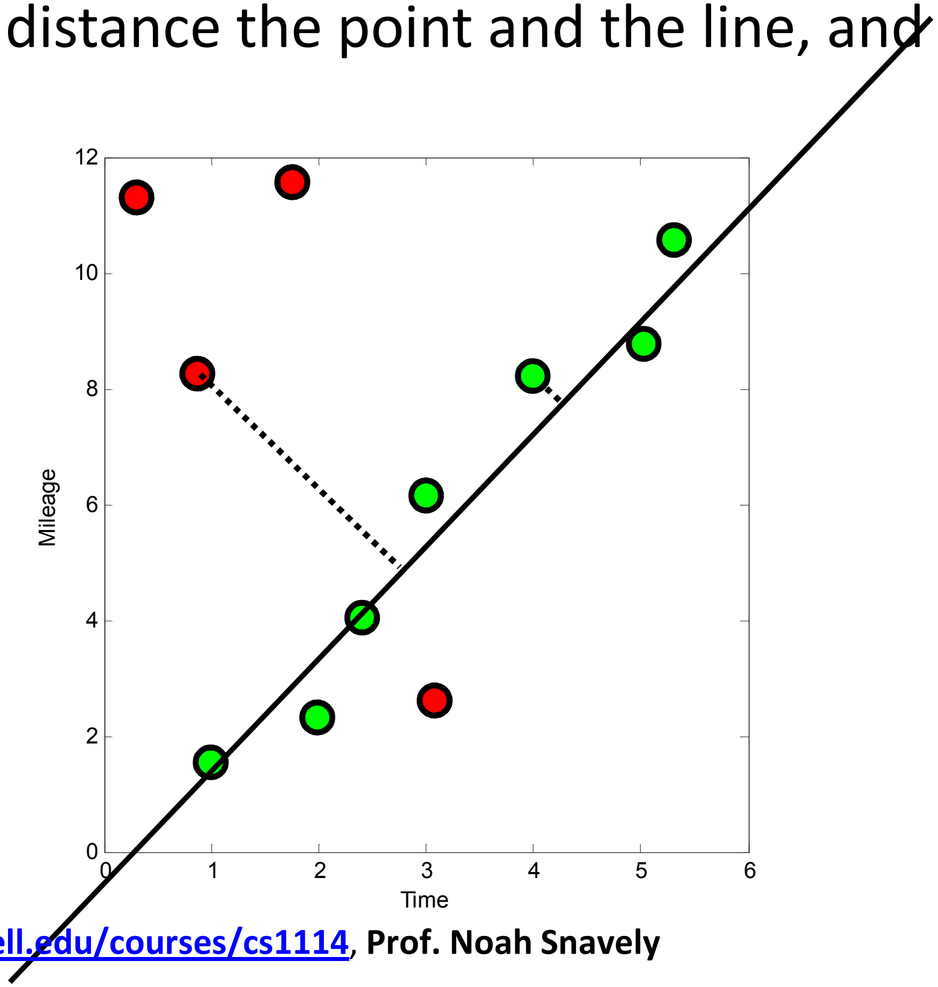
Linear regression vs Robust approaches



http://www.cs.cornell.edu/courses/cs1114, **Prof. Noah Snavely**

Testing goodness

- How can we tell if a point agrees with a line?
- Compute the distance the point and the line, and threshold



http://www.cs.cornell.edu/courses/cs1114, Prof. Noah Snavely

Testing goodness

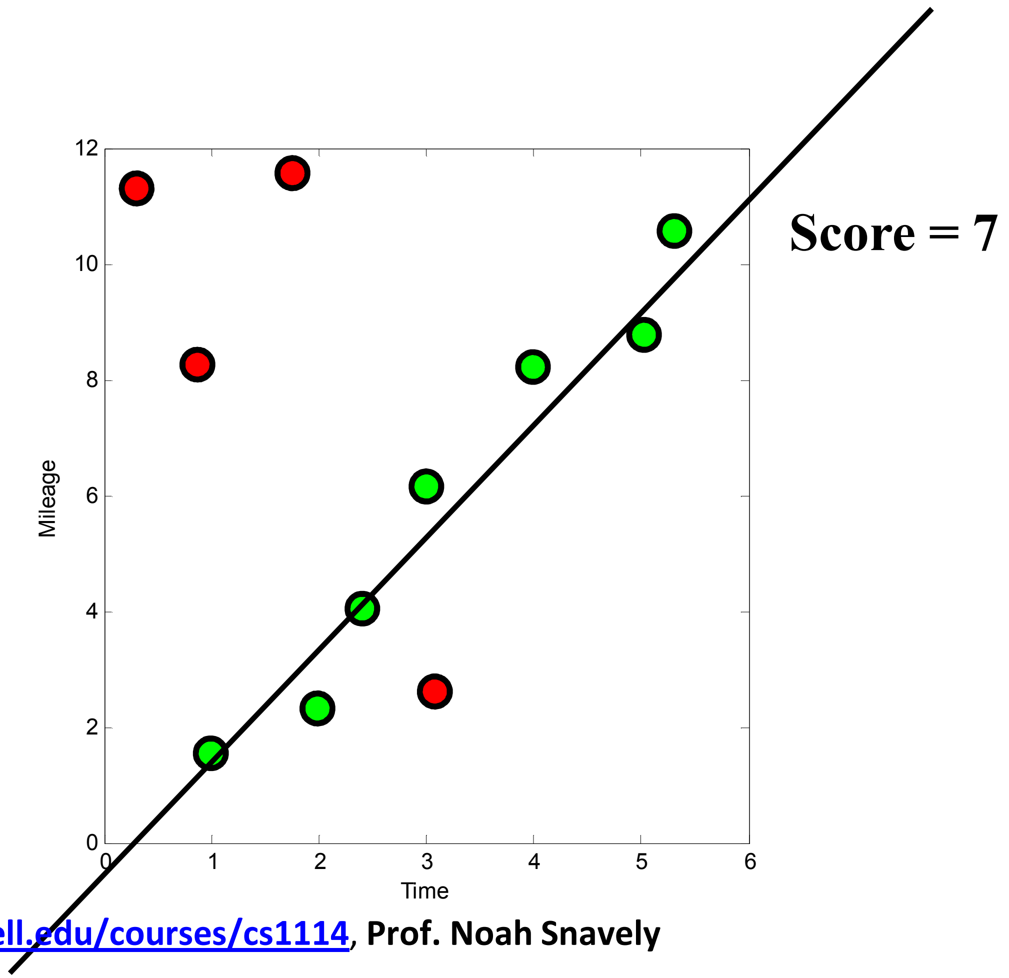
- If the distance is small, we call this point an *inlier* to the line
- If the distance is large, it's an *outlier* to the line
- For an inlier point and a good line, this distance will be close to (but not exactly) zero
- For an outlier point or bad line, this distance will probably be large

- Objective function: find the line with the most inliers (or the fewest outliers)

http://www.cs.cornell.edu/courses/cs1114, Prof. Noah Snavely

Optimizing for inlier count

- How do we find the best possible line?



http://www.cs.cornell.edu/courses/cs1114, Prof. Noah Snavely

RANSAC for estimating homography

- RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \boldsymbol{H} p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

http://www.cs.cornell.edu/courses/cs1114, Prof. Noah Snavely

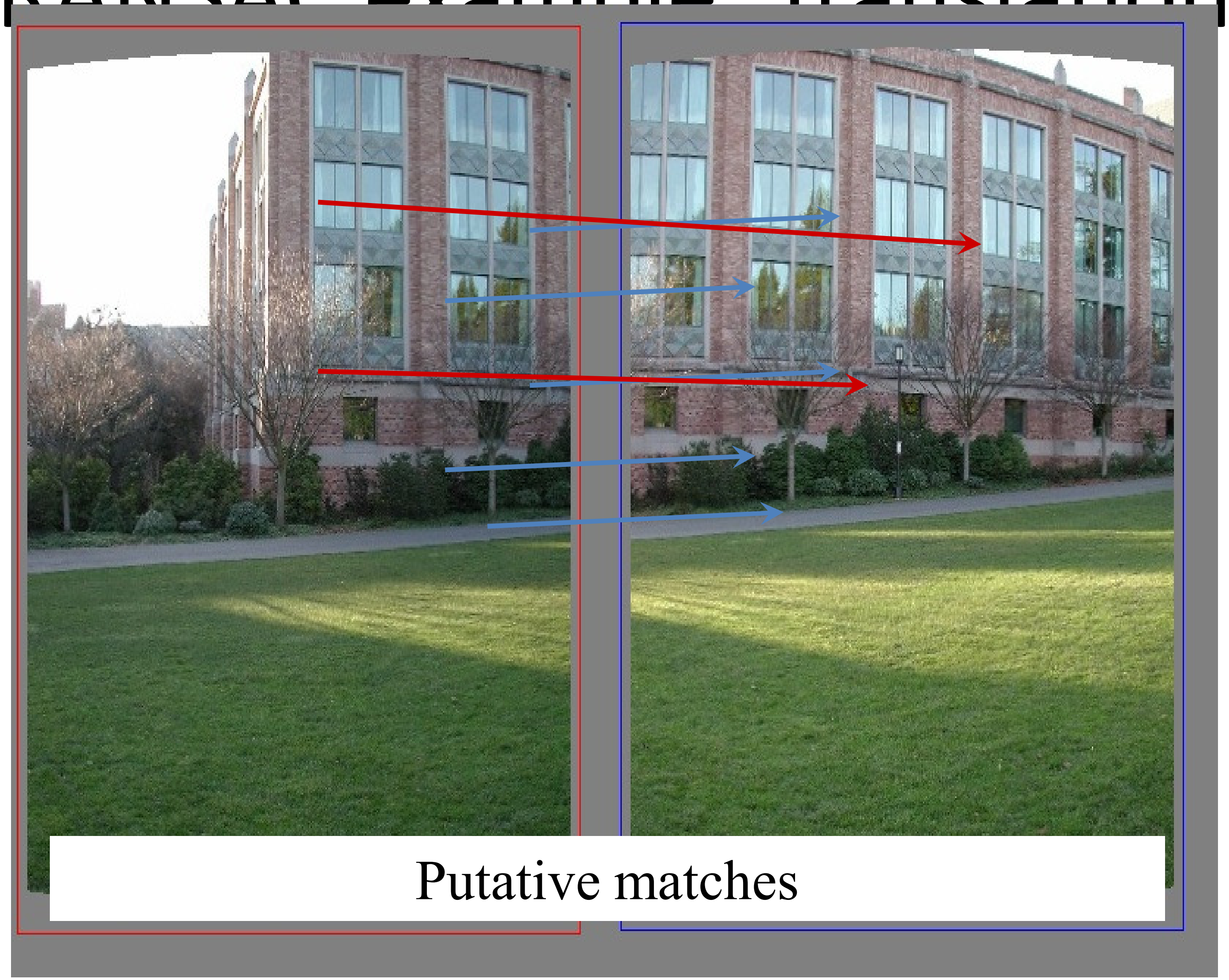
RANSAC

- RANSAC loop:

1. Randomly select a *seed group* of matches
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

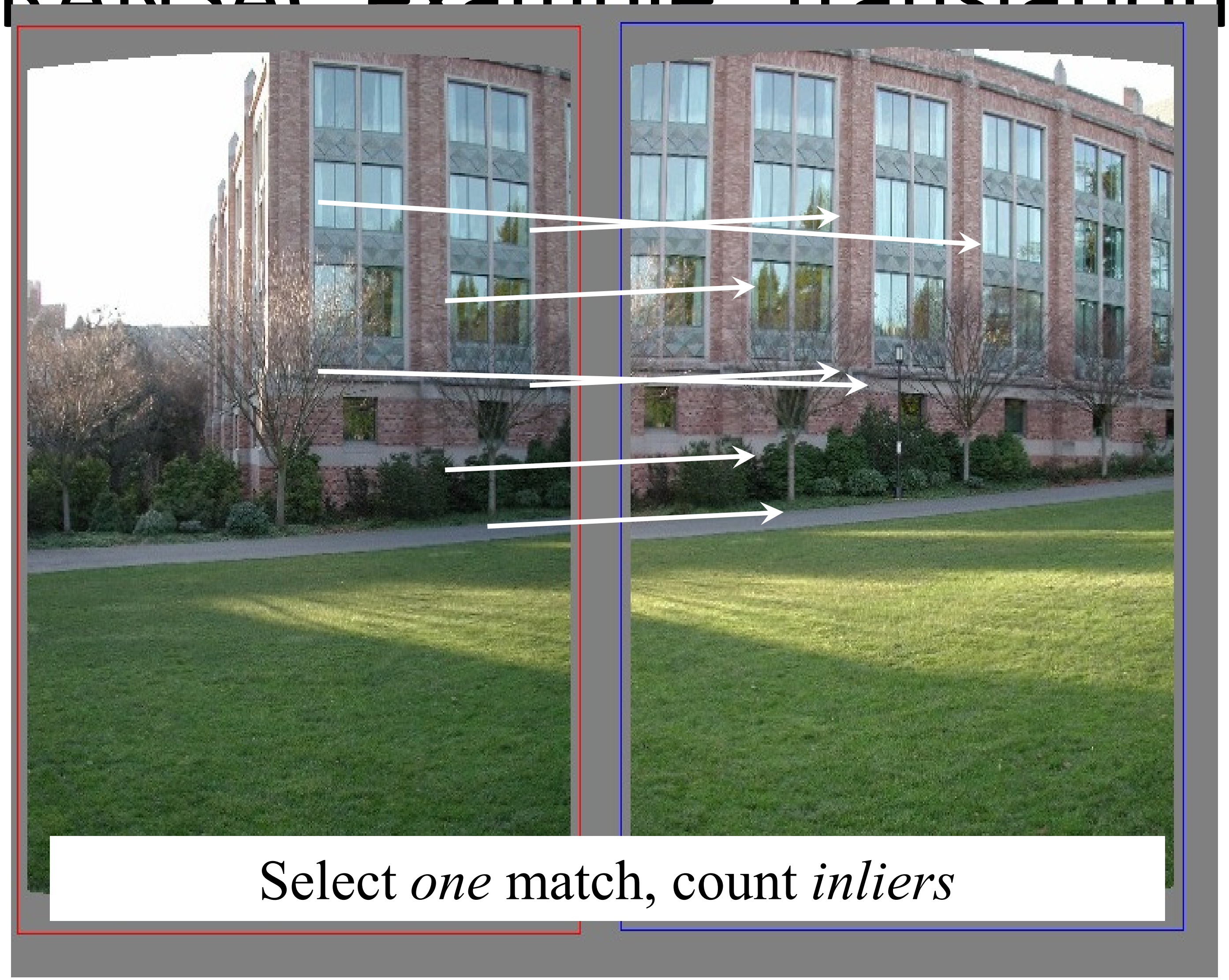
RANSAC example: Translation



15-463: Computational Photography
Alexei Efros, CMU, Fall 2005

Source: A. Efros

RANSAC example: Translation

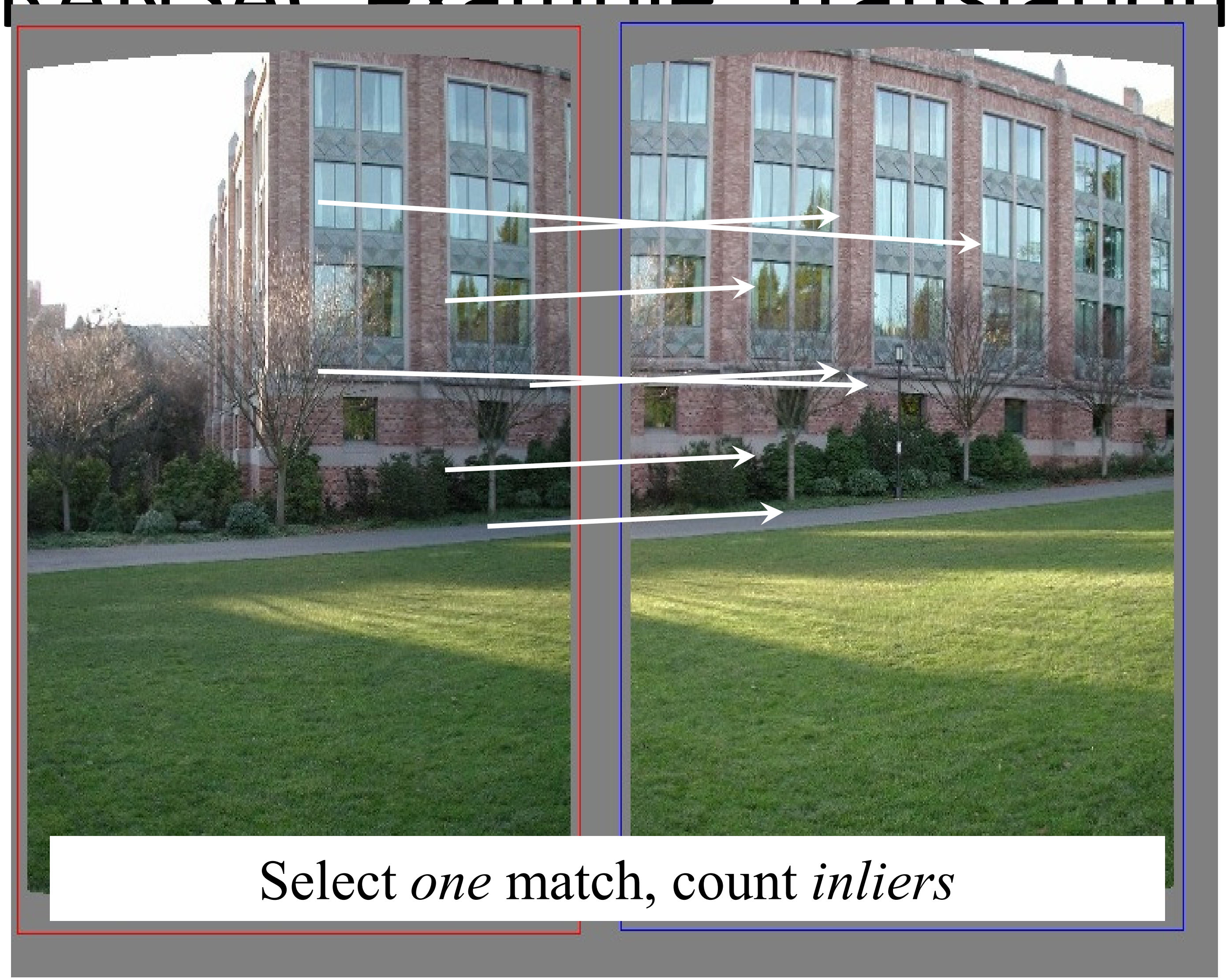


Select *one* match, count *inliers*

15-463: Computational Photography
Alexei Efros, CMU, Fall 2005

Source: A. Efros

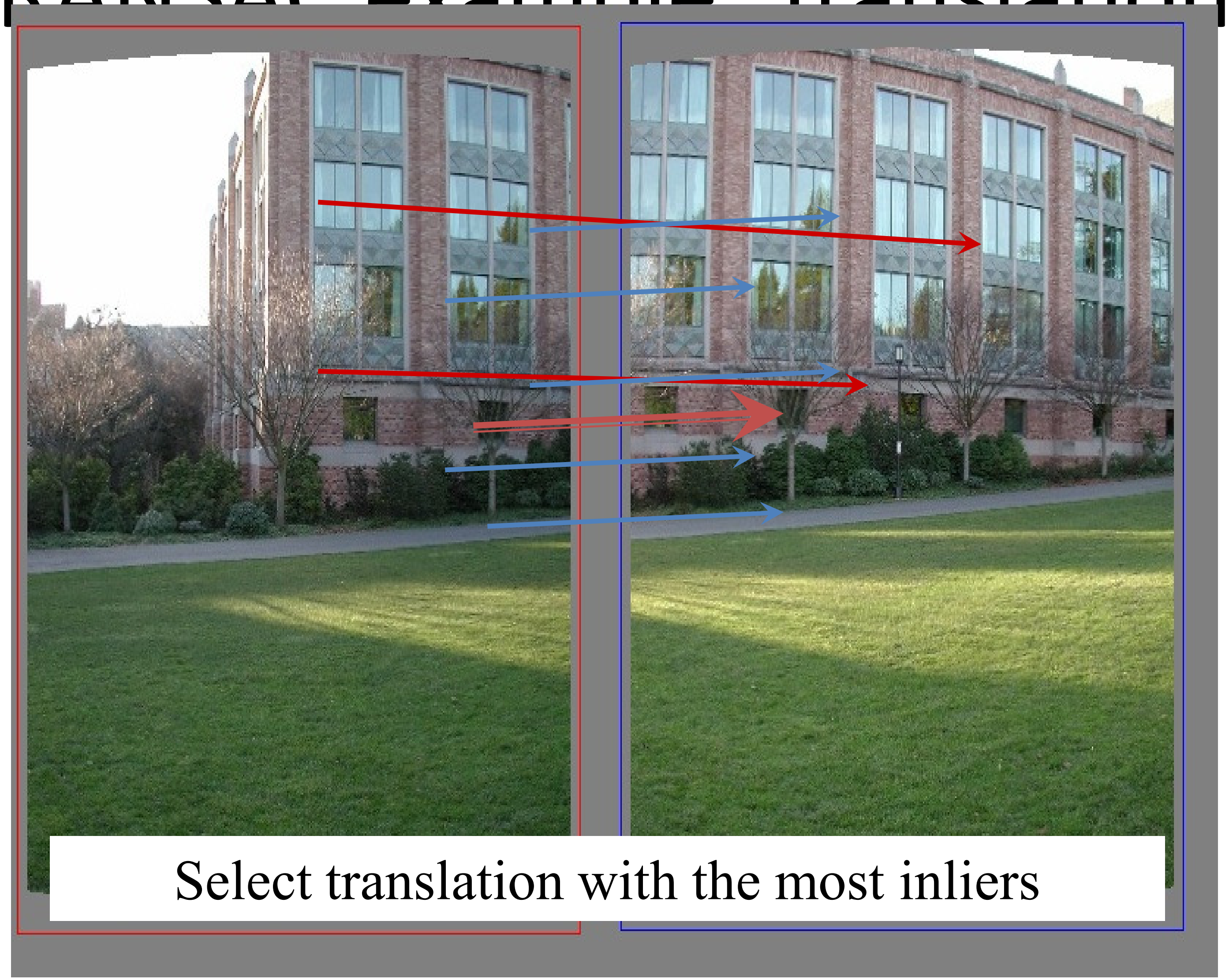
RANSAC example: Translation



15-463: Computational Photography
Alexei Efros, CMU, Fall 2005

Source: A. Efros

RANSAC example: Translation



Select translation with the most inliers

Source: A. Efros

How Many Samples?

On average

N	… number of point
I	… number of inliers
m	… size of the sample

P(good) = $$\frac{\binom{I}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1} \frac{I-j}{N-j}$$

mean time before the success

E(*k*) = 1 / P(good)

How Many Samples?

With confidence *p*

How large k?

... to hit at least one pair of points on the line l with probability larger than p (0.95)

Equivalently

... the probability of not hitting any pair of points on l is $\leq 1 - p$

How Many Samples?

N	… number of point
I	… number of inliers
m	… size of the sample

With confidence *p*

P(good) = $$\frac{\binom{I}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1} \frac{I-j}{N-j}$$

P(bad) = 1 – P(good)

P(bad *k* times) = $\left(1 - \text{P(good)}\right)^{k}$

How Many Samples?

With confidence p

$$\text{P(bad } k \text{ times)} = \left(1 - \text{P(good)}\right)^{k} \leq 1 - p$$

$$k \log\left(1 - \text{P(good)}\right) \leq \log(1 - p)$$

$$k \geq \log(1 - p) \,/\, \log\left(1 - \text{P(good)}\right)$$

How Many Samples

I / *N* [%]

Size of the sample *m*

	15%	20%	30%	40%	50%	70%
2	132	73	32	17	10	4
4	5916	1871	368	116	46	11
7	$1.75 \cdot 10^6$	$2.34 \cdot 10^5$	$1.37 \cdot 10^4$	1827	382	35
8	$1.17 \cdot 10^7$	$1.17 \cdot 10^6$	$4.57 \cdot 10^4$	4570	765	50
12	$2.31 \cdot 10^{10}$	$7.31 \cdot 10^8$	$5.64 \cdot 10^6$	$1.79 \cdot 10^5$	$1.23 \cdot 10^4$	215
18	$2.08 \cdot 10^{15}$	$1.14 \cdot 10^{13}$	$7.73 \cdot 10^9$	$4.36 \cdot 10^7$	$7.85 \cdot 10^5$	1838
30	∞	∞	$1.35 \cdot 10^{16}$	$2.60 \cdot 10^{12}$	$3.22 \cdot 10^9$	$1.33 \cdot 10^5$
40	∞	∞	∞	$2.70 \cdot 10^{16}$	$3.29 \cdot 10^{12}$	$4.71 \cdot 10^6$