Basic registration

D.A. Forsyth

University of Illinois at Urbana Champaign

General frameworks

- Register two sets of points
  - where correspondence is known exactly
    - eg barcode, etc. reference points
  - where correspondence is estimated, but quite well
    - eg two images, interest points
  - where correspondence might be hard to estimate
    - but registration is possible
    - eg two lidar images of about the same stuff

Setup for affine tx

For an affine transformation, $\mathcal{T}(\mathbf{y})$ is $\mathcal{M}\mathbf{y}+\mathbf{t}$. Further, there is a transformation $\mathcal{T}$ so that $\mathcal{T}(\mathbf{y}_i)$ is close to $\mathbf{x}_i$ for each i. Write $\mathbf{r}_i$ for the vector from the transformed $\mathbf{y}_i$ to $\mathbf{x}_i$, so

$$\mathbf{r}_i(\mathcal{M},\mathbf{t}) = (\mathbf{x}_i - (\mathcal{M}\mathbf{y}_i + \mathbf{t}))$$

and

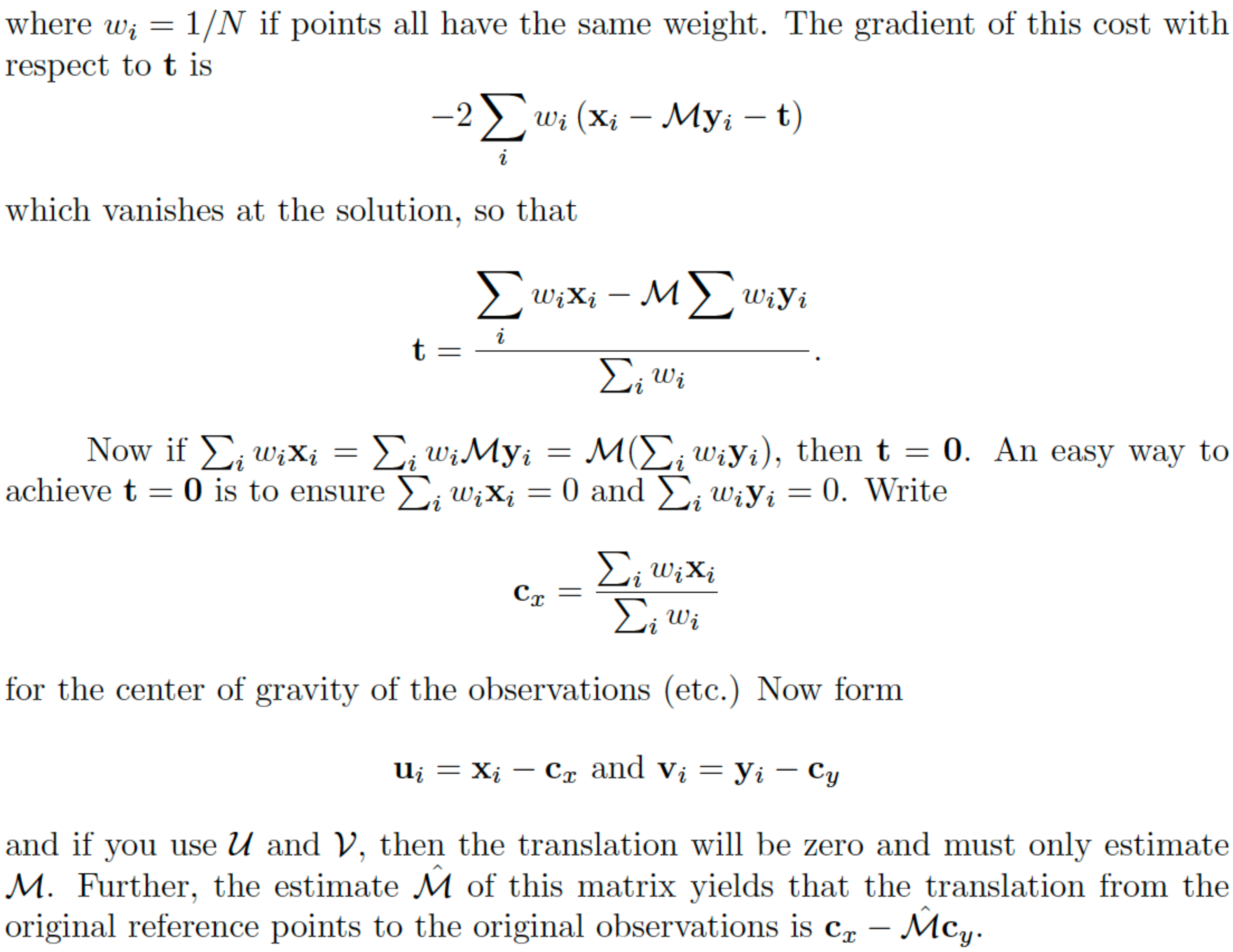
$$C_u(\mathcal{M},\mathbf{t}) = (1/N)\sum_i \mathbf{r}_i^T\mathbf{r}_i$$

should be small. Because it will be useful later, assume that there is a weight w_i for each pair and work with

$$C(\mathcal{M},\mathbf{t}) = \sum_i w_i\mathbf{r}_i^T\mathbf{r}_i$$

where $w_i = 1/N$ if points all have the same weight. The gradient of this cost with

Translation term



where $w_i = 1/N$ if points all have the same weight. The gradient of this cost with respect to $\mathbf{t}$ is

$$-2\sum_i w_i\left(\mathbf{x}_i - \mathcal{M}\mathbf{y}_i - \mathbf{t}\right)$$

which vanishes at the solution, so that

$$\mathbf{t} = \frac{\sum\limits_i w_i\mathbf{x}_i - \mathcal{M}\sum w_i\mathbf{y}_i}{\sum_i w_i}.$$

Now if $\sum_i w_i\mathbf{x}_i = \sum_i w_i\mathcal{M}\mathbf{y}_i = \mathcal{M}(\sum_i w_i\mathbf{y}_i)$, then $\mathbf{t} = \mathbf{0}$. An easy way to achieve $\mathbf{t} = \mathbf{0}$ is to ensure $\sum_i w_i\mathbf{x}_i = 0$ and $\sum_i w_i\mathbf{y}_i = 0$. Write

$$\mathbf{c}_x = \frac{\sum_i w_i\mathbf{x}_i}{\sum_i w_i}$$

for the center of gravity of the observations (etc.) Now form

$$\mathbf{u}_i = \mathbf{x}_i - \mathbf{c}_x \text{ and } \mathbf{v}_i = \mathbf{y}_i - \mathbf{c}_y$$

and if you use $\mathcal{U}$ and $\mathcal{V}$, then the translation will be zero and must only estimate $\mathcal{M}$. Further, the estimate $\hat{\mathcal{M}}$ of this matrix yields that the translation from the original reference points to the original observations is $\mathbf{c}_x - \hat{\mathcal{M}}\mathbf{c}_y$.

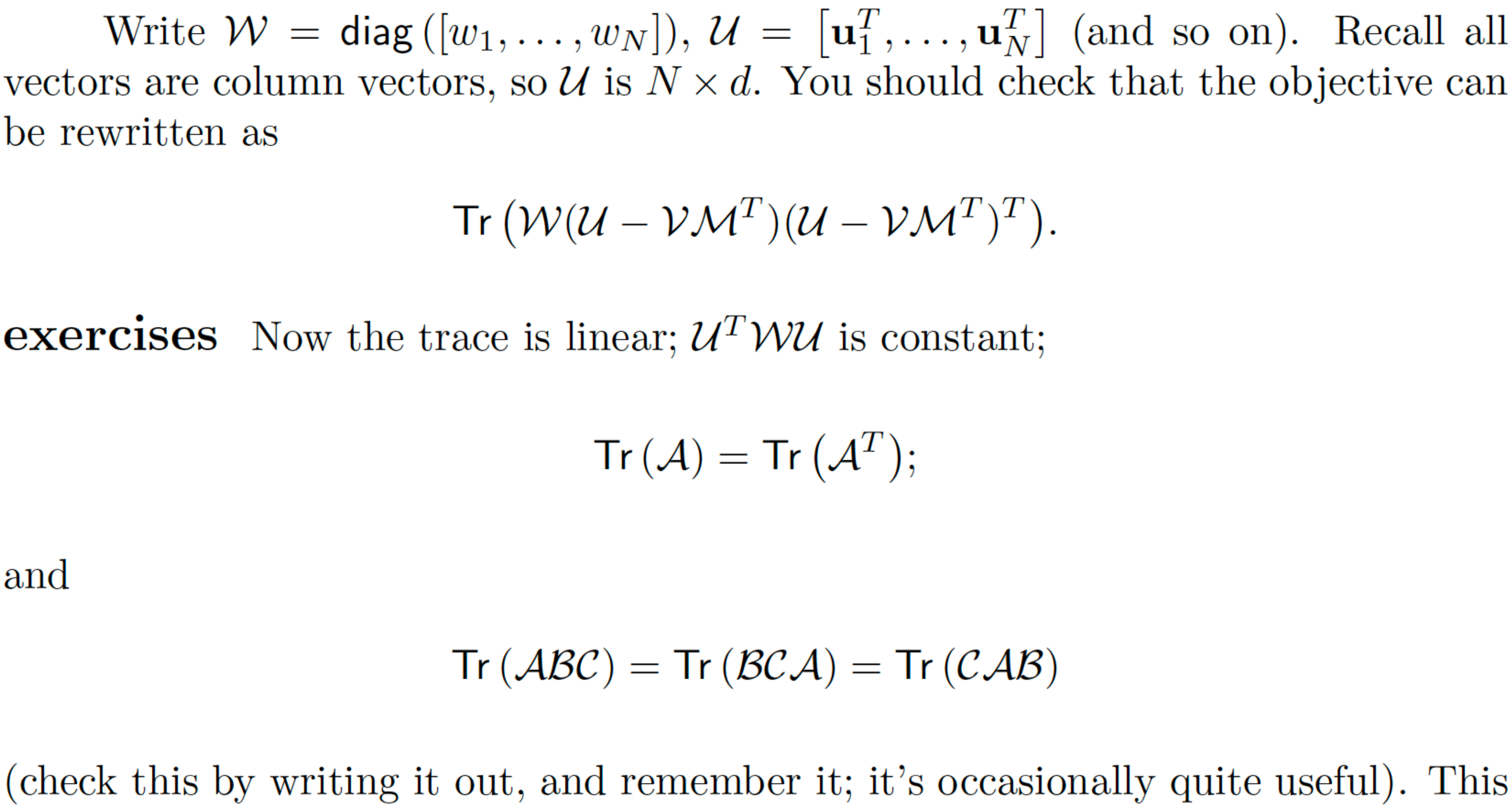
Finding M (compact form)

Finding $\mathcal{M}$ now reduces to minimizing

$$\sum_i w_i \left(\mathbf{u}_i - \mathcal{M}\mathbf{v}_i\right)^T \left(\mathbf{u}_i - \mathcal{M}\mathbf{v}_i\right)$$

as a function of $\mathcal{M}$. The natural procedure – take a derivative and set to zero, and obtain a linear system (**exercises**) – works fine, but it is helpful to apply some compact and decorative notation.

Finding M (long form)



Write $\mathcal{W} = \mathsf{diag}\left([w_1, \ldots, w_N]\right)$, $\mathcal{U} = \left[\mathbf{u}_1^T, \ldots, \mathbf{u}_N^T\right]$ (and so on). Recall all vectors are column vectors, so $\mathcal{U}$ is $N \times d$. You should check that the objective can be rewritten as

$$\mathsf{Tr}\left(\mathcal{W}(\mathcal{U} - \mathcal{V}\mathcal{M}^T)(\mathcal{U} - \mathcal{V}\mathcal{M}^T)^T\right).$$

exercises Now the trace is linear; $\mathcal{U}^T\mathcal{W}\mathcal{U}$ is constant;

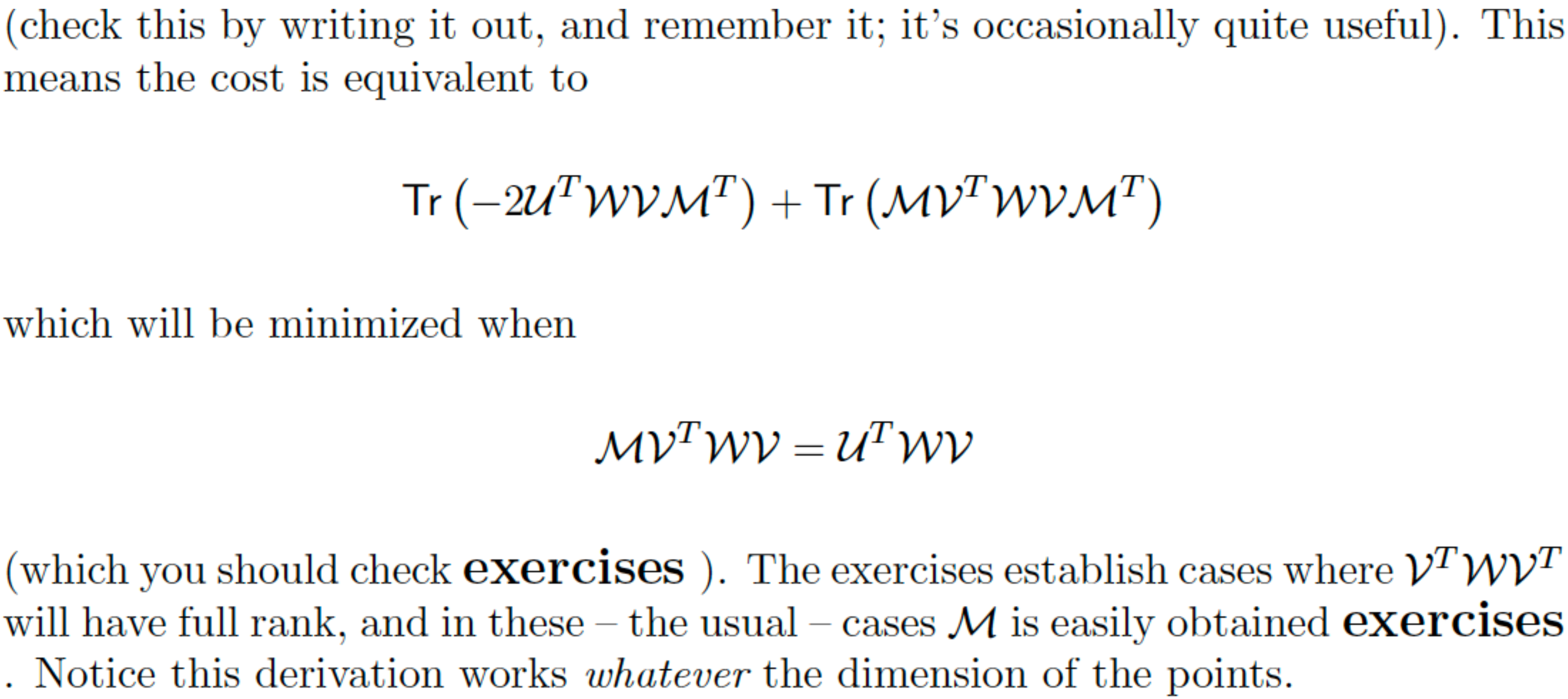
$$\mathsf{Tr}\left(\mathcal{A}\right) = \mathsf{Tr}\left(\mathcal{A}^T\right);$$

and

$$\mathsf{Tr}\left(\mathcal{A}\mathcal{B}\mathcal{C}\right) = \mathsf{Tr}\left(\mathcal{B}\mathcal{C}\mathcal{A}\right) = \mathsf{Tr}\left(\mathcal{C}\mathcal{A}\mathcal{B}\right)$$

(check this by writing it out, and remember it; it's occasionally quite useful). This

Finding M (long form)



(check this by writing it out, and remember it; it's occasionally quite useful). This means the cost is equivalent to

$$\mathsf{Tr}\left(-2\mathcal{U}^T\mathcal{W}\mathcal{V}\mathcal{M}^T\right)+\mathsf{Tr}\left(\mathcal{M}\mathcal{V}^T\mathcal{W}\mathcal{V}\mathcal{M}^T\right)$$

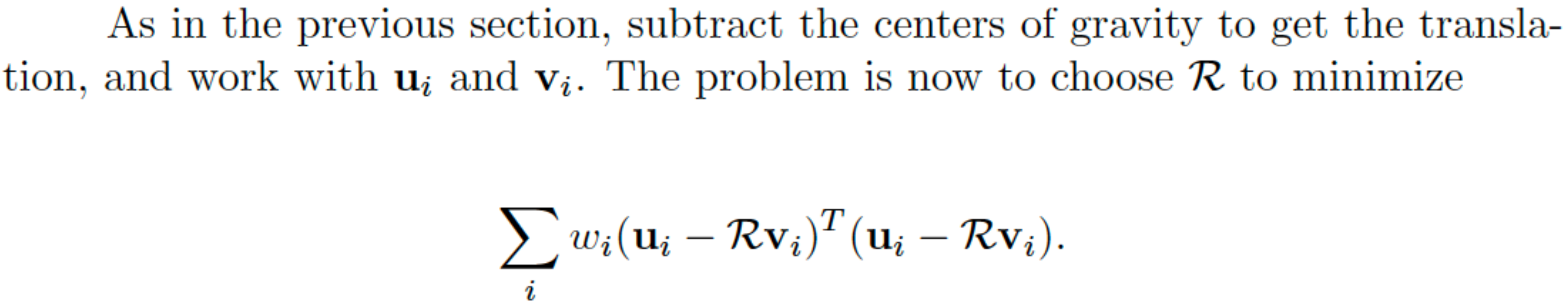
which will be minimized when

$$\mathcal{M}\mathcal{V}^T\mathcal{W}\mathcal{V}=\mathcal{U}^T\mathcal{W}\mathcal{V}$$

(which you should check **exercises**). The exercises establish cases where $\mathcal{V}^T\mathcal{W}\mathcal{V}^T$ will have full rank, and in these – the usual – cases $\mathcal{M}$ is easily obtained **exercises** . Notice this derivation works *whatever* the dimension of the points.

Euclidean motion

- Most interesting in 2D or 3D
- The matrix is a rotation matrix
- You can do this in closed form (not widely known)



As in the previous section, subtract the centers of gravity to get the translation, and work with $\mathbf{u}_i$ and $\mathbf{v}_i$. The problem is now to choose $\mathcal{R}$ to minimize

$$\sum_i w_i(\mathbf{u}_i - \mathcal{R}\mathbf{v}_i)^T(\mathbf{u}_i - \mathcal{R}\mathbf{v}_i).$$

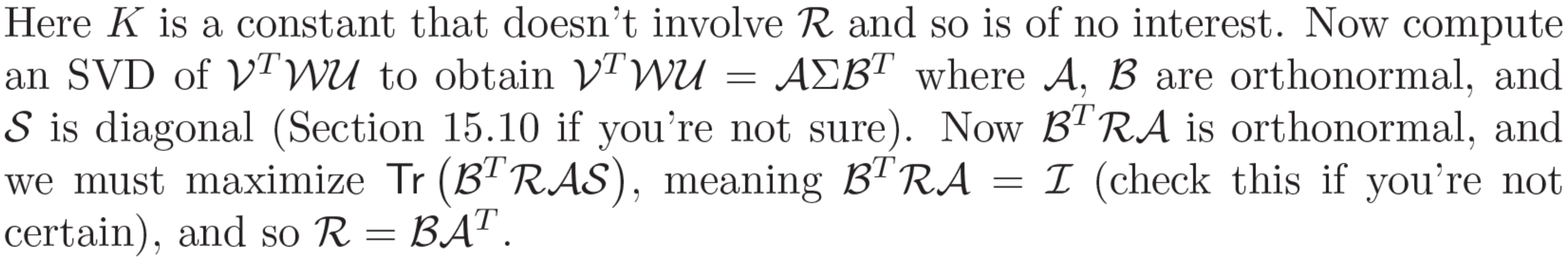
Euclidean motion

$$\begin{aligned}\sum_i w_i(\mathbf{u}_i - \mathcal{R}\mathbf{v}_i)^T(\mathbf{u}_i - \mathcal{R}\mathbf{v}_i) &= \mathsf{Tr}\left(\mathcal{W}(\mathcal{U} - \mathcal{V}\mathcal{R}^T)(\mathcal{U} - \mathcal{V}\mathcal{R})^T\right) \\ &= \mathsf{Tr}\left(-2\mathcal{V}^T\mathcal{W}\mathcal{U}\mathcal{R}\right) + K \\ &\quad \text{(because } \mathcal{R}^T\mathcal{R} = \mathcal{I}\text{)}\end{aligned}$$

Terms not involving R, so of no interest

Euclidean motion

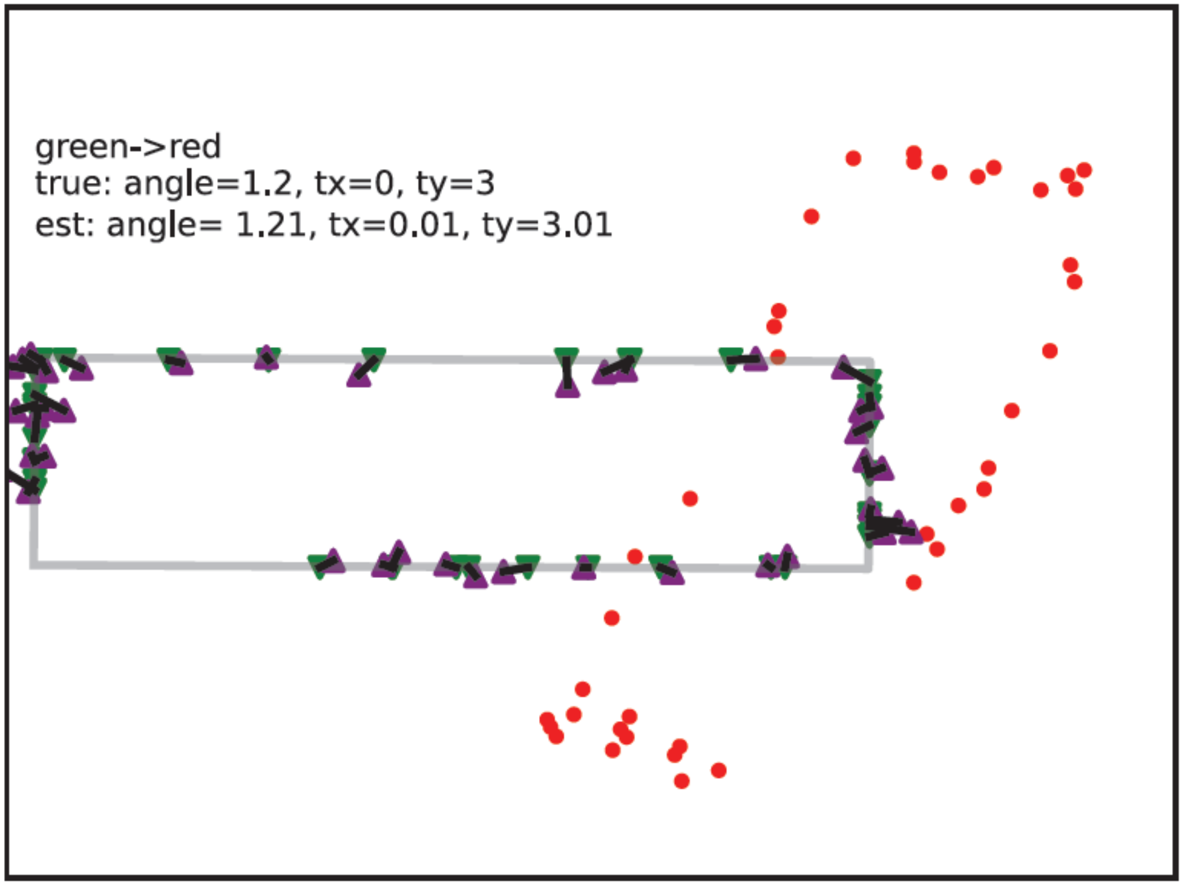
$$\mathsf{Tr}\left(-2\mathcal{V}^T\mathcal{W}\mathcal{U}\mathcal{R}\right)$$



Here K is a constant that doesn't involve $\mathcal{R}$ and so is of no interest. Now compute an SVD of $\mathcal{V}^T\mathcal{W}\mathcal{U}$ to obtain $\mathcal{V}^T\mathcal{W}\mathcal{U} = \mathcal{A}\Sigma\mathcal{B}^T$ where $\mathcal{A}$, $\mathcal{B}$ are orthonormal, and $\mathcal{S}$ is diagonal (Section 15.10 if you're not sure). Now $\mathcal{B}^T\mathcal{R}\mathcal{A}$ is orthonormal, and we must maximize $\mathsf{Tr}\left(\mathcal{B}^T\mathcal{R}\mathcal{A}\mathcal{S}\right)$, meaning $\mathcal{B}^T\mathcal{R}\mathcal{A} = \mathcal{I}$ (check this if you're not certain), and so $\mathcal{R} = \mathcal{B}\mathcal{A}^T$.

Euclidean est. well behaved under noise

0.05

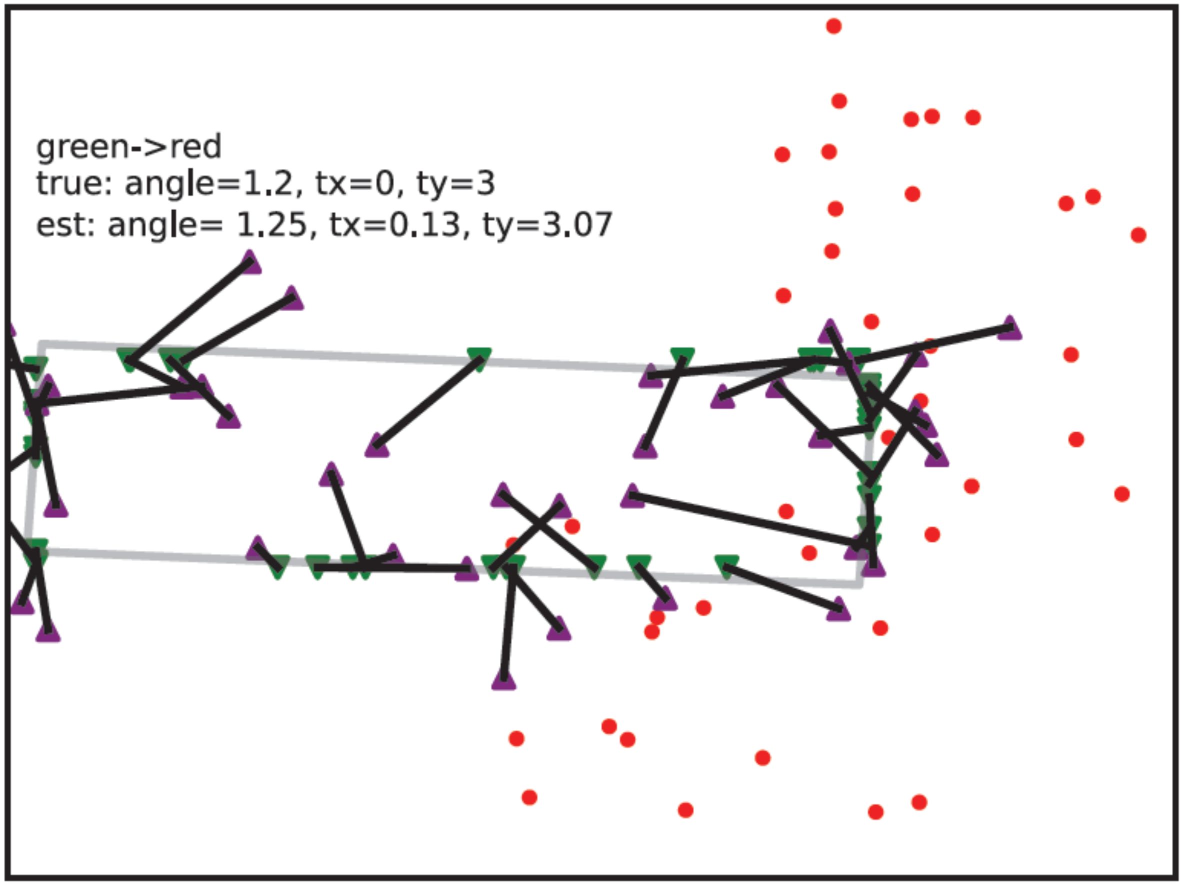


- Green triangles – target pts lying on gray rectangle
- Red dots – source
- (target points transformed, then noise added)
- Purple triangles – apply estimated transformation to red points
- Gray rectangle – transformation applied to true rectangle underlying red points

- Notice:
- transformation is about right, not massively disrupted by noise

Euclidean est. well behaved under noise

0.3



- Green triangles – target pts lying on gray rectangle
- Red dots – source
- (target points transformed, then noise added)
- Purple triangles – apply estimated transformation to red points
- Gray rectangle – transformation applied to true rectangle underlying red points
- Notice:
- transformation is about right, only somewhat disrupted by noise

Projective transformations

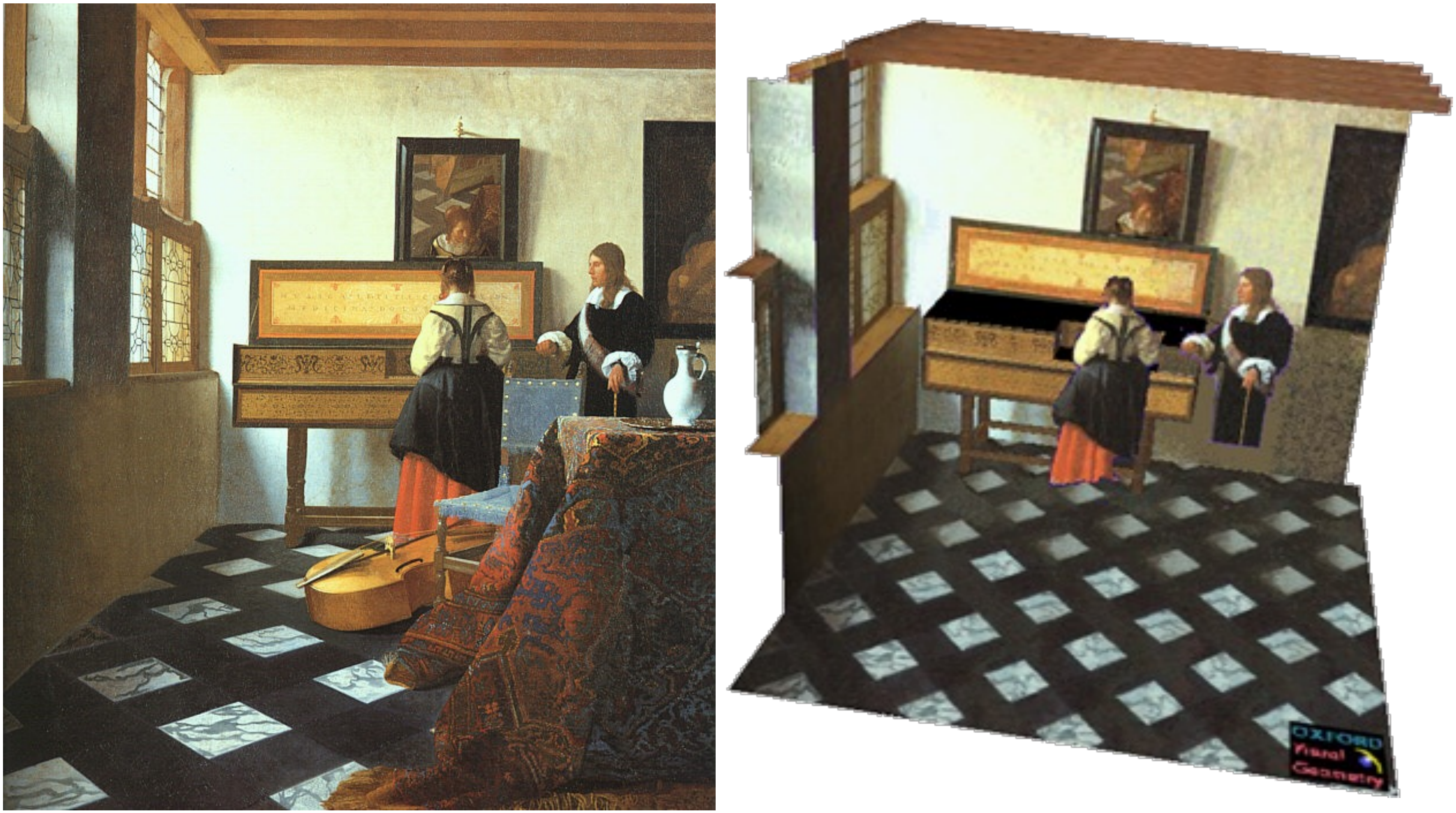
- In 2D

$$\begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{m_{11}y_1 + m_{12}y_2 + m_{13}}{m_{31}y_1 + m_{32}y_2 + m_{33}} \\ \\ \frac{m_{21}y_1 + m_{22}y_2 + m_{23}}{m_{31}y_1 + m_{32}y_y + m_{33}} \end{bmatrix} = \mathcal{M}(\mathbf{y}).$$

Homographies == Projective tx

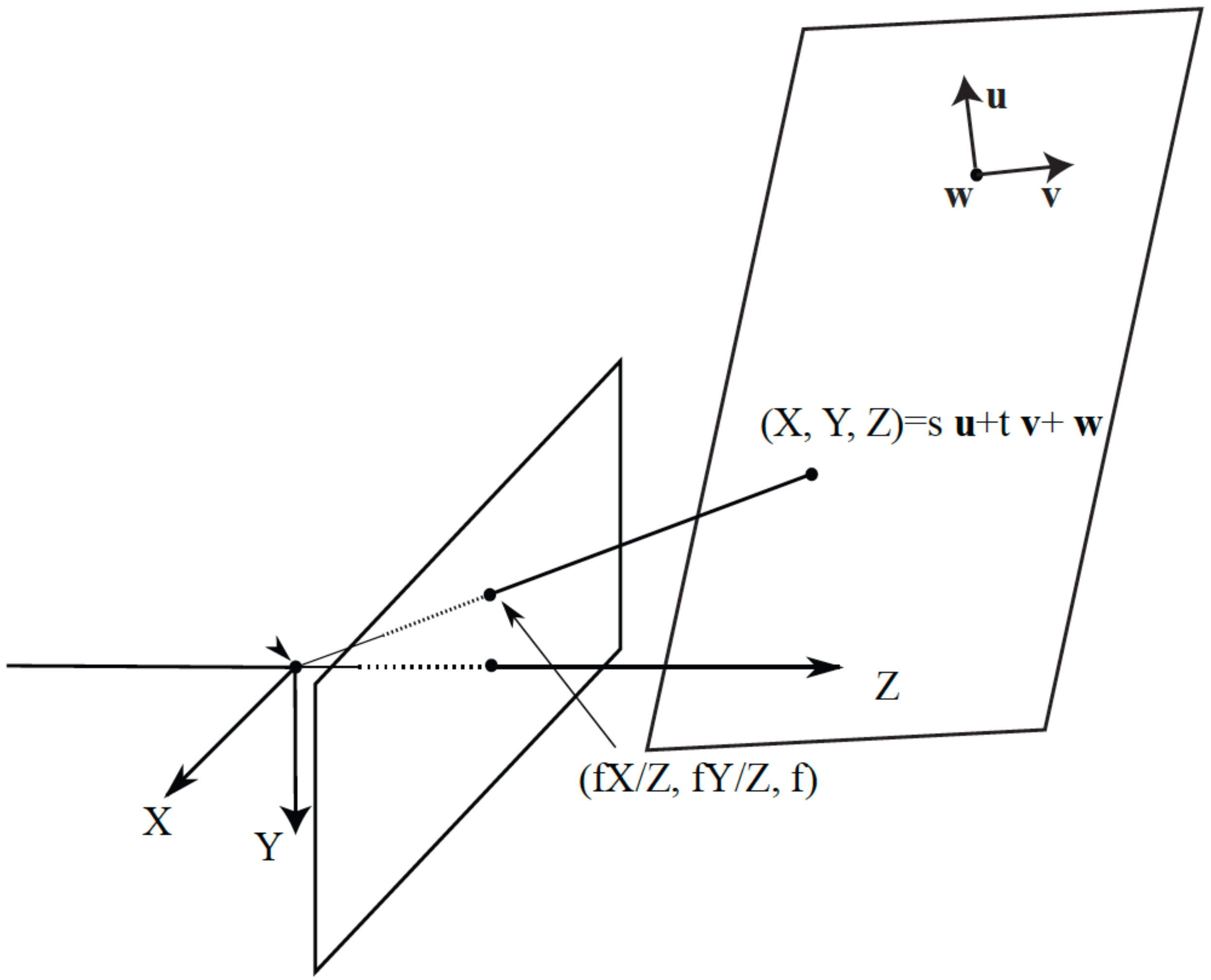
- The mapping from a pattern on a plane in 3D to a pattern in an image is a projective transformation

- This means you can rectify by registration
  - eg see what you know to be a square in image
  - use vertices to register
  - recover pattern

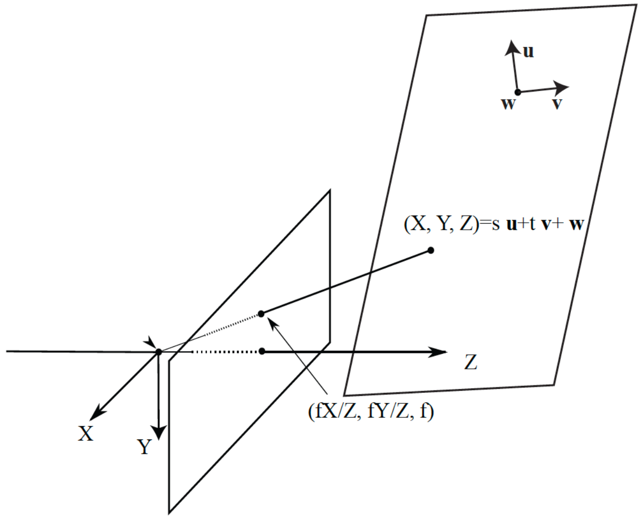


A. Criminisi et al. Bringing Pictorial Space to Life: computer techniques for the analysis of paintings,
Proc. Computers and the History of Art, 2002

Homographies == Projective tx



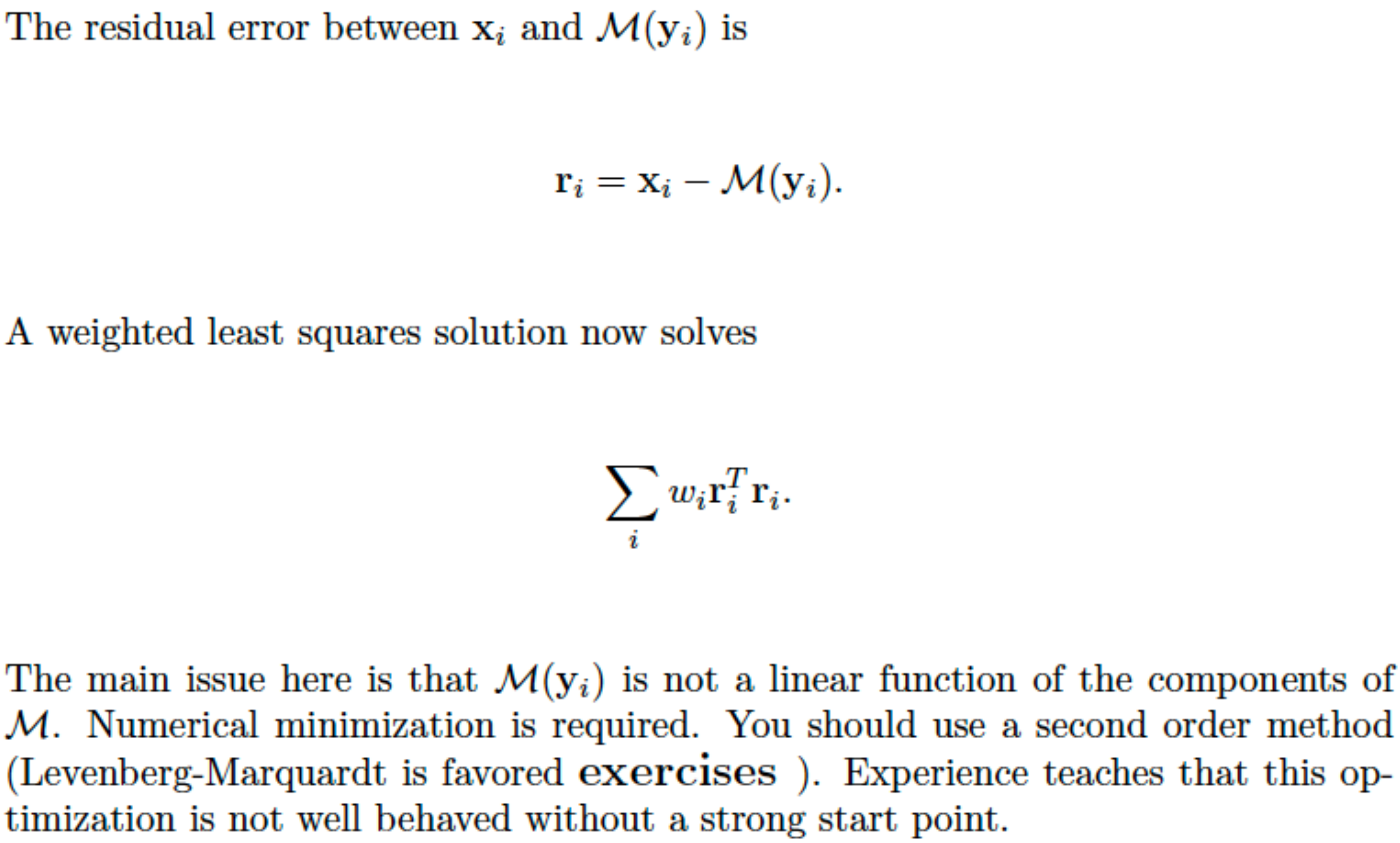
Homographies == Projective tx



The coordinate system on the plane is $(s,t)^T$, and the points on the plane in 3D are parametrized by $s\mathbf{u}+t\mathbf{v}+\mathbf{w}$, where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are vectors in 3D and $\mathbf{u}$, $\mathbf{v}$ are not parallel. Recall from Section **??** the geometric model that the pinhole camera maps the point $(X,Y,Z)^T$ in 3D to the point $(fX/Z, fY/Z)^T$ on the image plane. In turn, the point $(s,t)^T$ on the plane maps to

$$\left[\begin{array}{c} f\frac{su_1+tv_1+w_1}{su_3+tv_3+w_3} \\ f\frac{su_2+tv_2+w_2}{su_3+tv_3+w_3} \end{array}\right]$$

Minimizing the residual



The residual error between $\mathbf{x}_i$ and $\mathcal{M}(\mathbf{y}_i)$ is

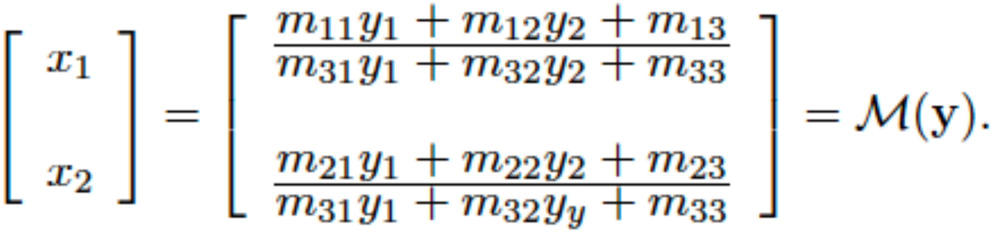
$$\mathbf{r}_i = \mathbf{x}_i - \mathcal{M}(\mathbf{y}_i).$$

A weighted least squares solution now solves

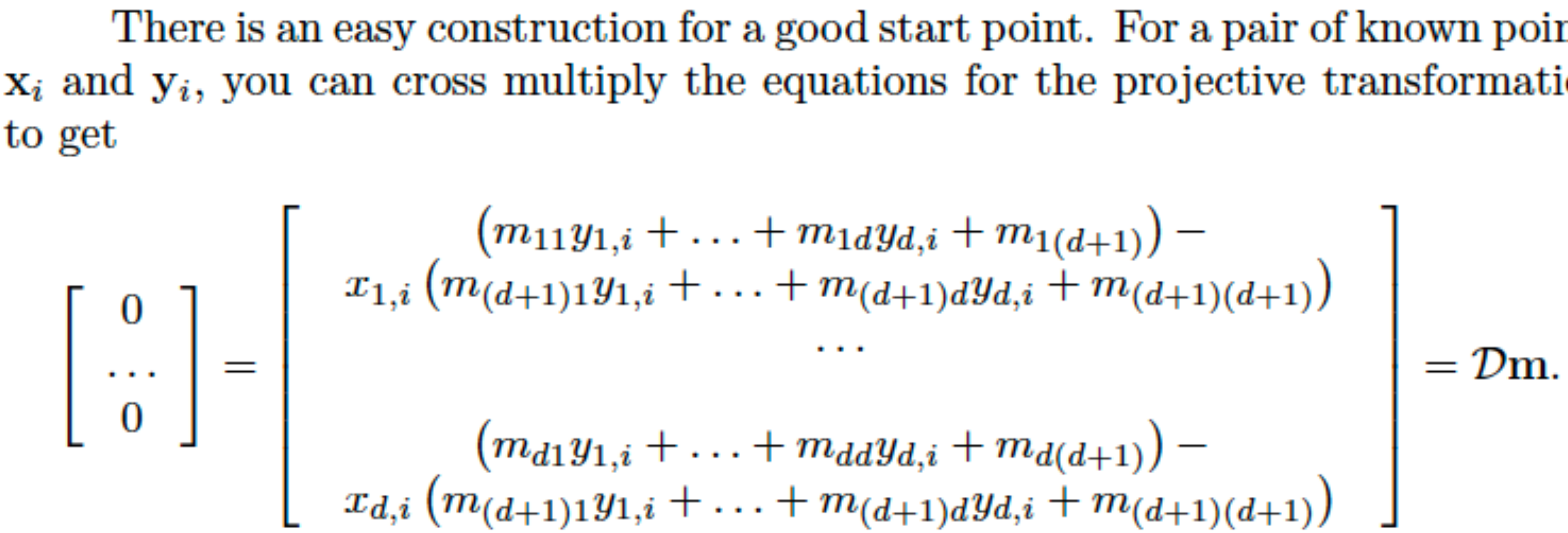
$$\sum_i w_i \mathbf{r}_i^T \mathbf{r}_i.$$

The main issue here is that $\mathcal{M}(\mathbf{y}_i)$ is not a linear function of the components of $\mathcal{M}$. Numerical minimization is required. You should use a second order method (Levenberg-Marquardt is favored exercises). Experience teaches that this optimization is not well behaved without a strong start point.

Getting a start point



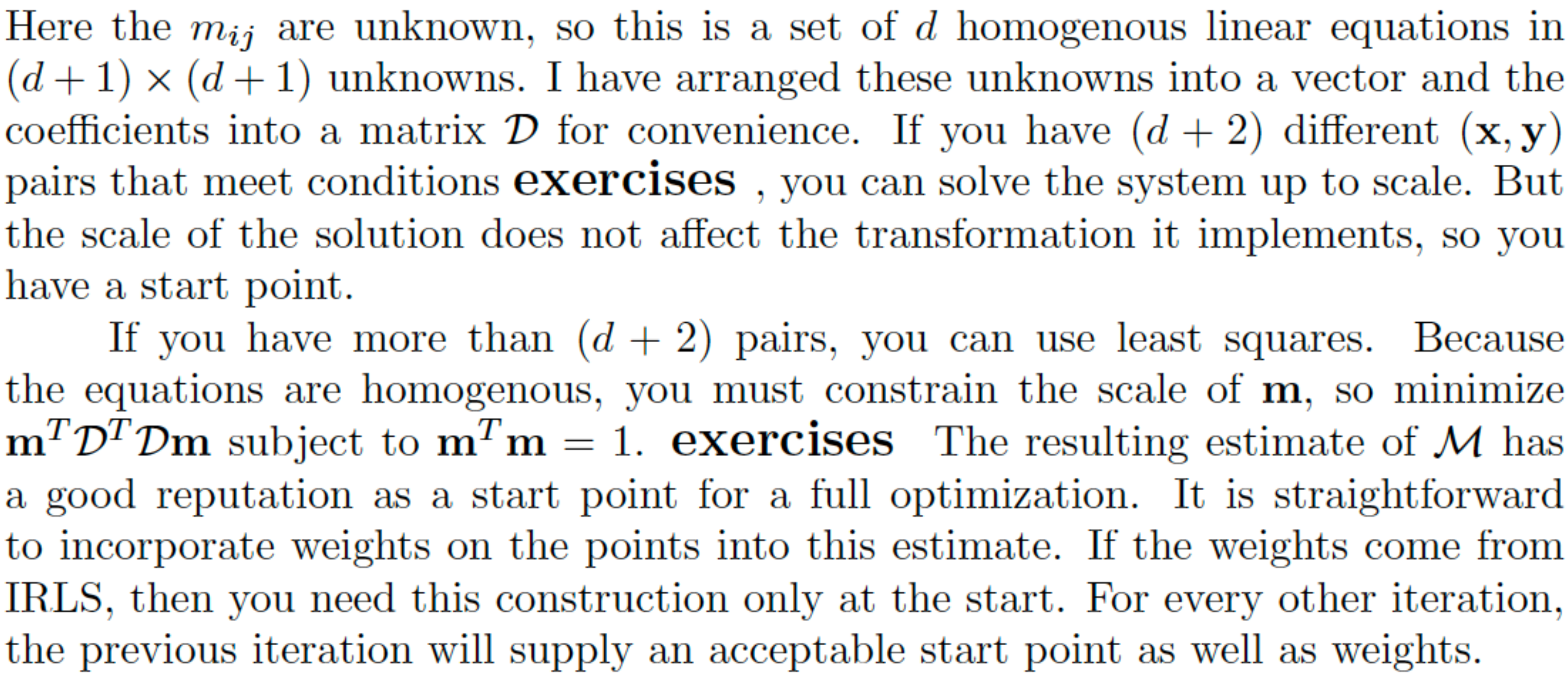
$$\begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{m_{11}y_1 + m_{12}y_2 + m_{13}}{m_{31}y_1 + m_{32}y_2 + m_{33}} \\ \\ \frac{m_{21}y_1 + m_{22}y_2 + m_{23}}{m_{31}y_1 + m_{32}y_y + m_{33}} \end{bmatrix} = \mathcal{M}(\mathbf{y}).$$



There is an easy construction for a good start point. For a pair of known points $\mathbf{x}_i$ and $\mathbf{y}_i$, you can cross multiply the equations for the projective transformation to get

$$\begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} = \begin{bmatrix} (m_{11}y_{1,i} + \ldots + m_{1d}y_{d,i} + m_{1(d+1)}) - \\ x_{1,i}\left(m_{(d+1)1}y_{1,i} + \ldots + m_{(d+1)d}y_{d,i} + m_{(d+1)(d+1)}\right) \\ \dots \\ \\ (m_{d1}y_{1,i} + \ldots + m_{dd}y_{d,i} + m_{d(d+1)}) - \\ x_{d,i}\left(m_{(d+1)1}y_{1,i} + \ldots + m_{(d+1)d}y_{d,i} + m_{(d+1)(d+1)}\right) \end{bmatrix} = \mathcal{D}\mathbf{m}.$$

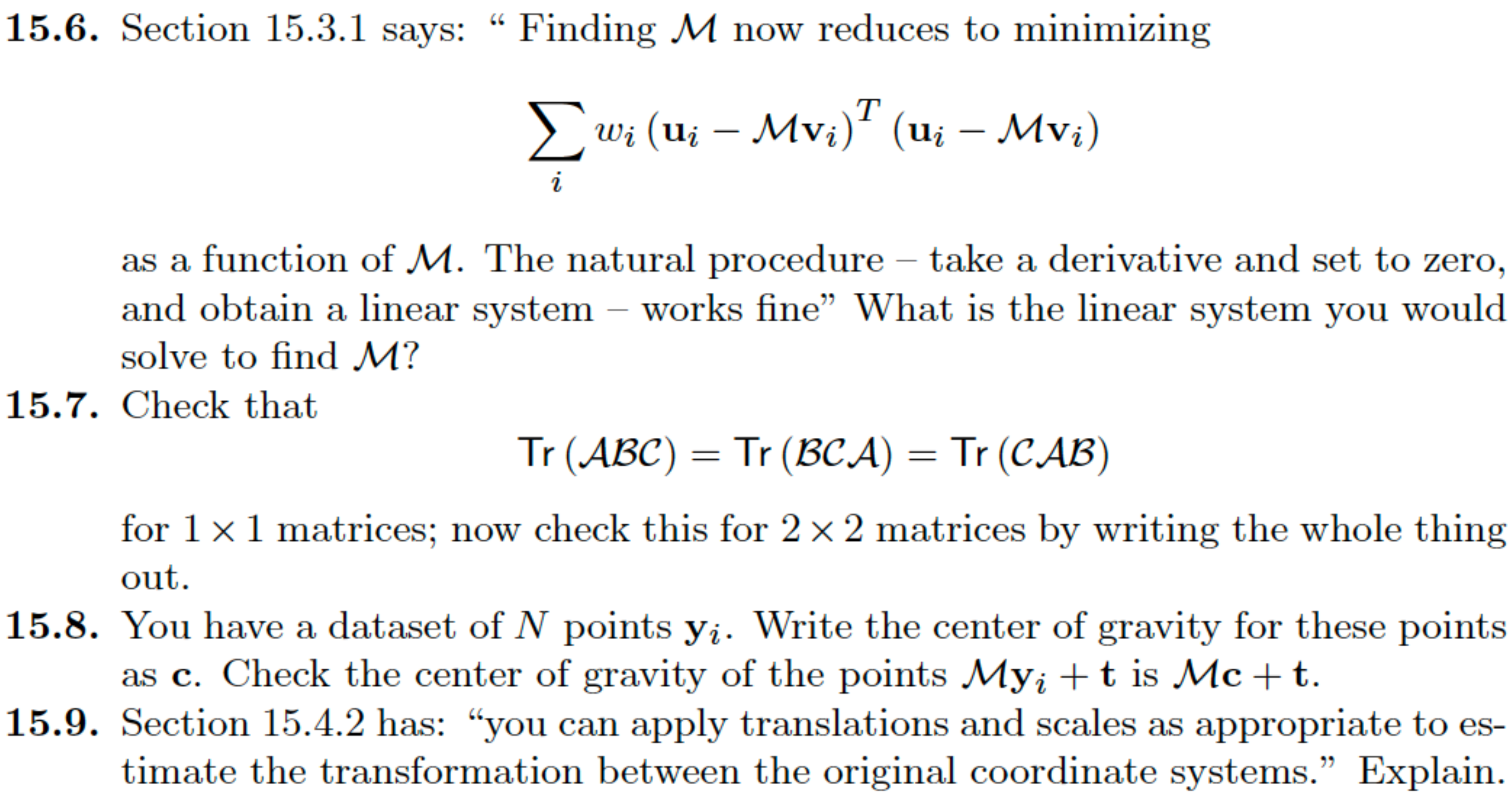
Getting a start point



Here the m_{ij} are unknown, so this is a set of d homogenous linear equations in $(d+1) \times (d+1)$ unknowns. I have arranged these unknowns into a vector and the coefficients into a matrix $\mathcal{D}$ for convenience. If you have $(d+2)$ different $(\mathbf{x}, \mathbf{y})$ pairs that meet conditions **exercises** , you can solve the system up to scale. But the scale of the solution does not affect the transformation it implements, so you have a start point.

If you have more than $(d+2)$ pairs, you can use least squares. Because the equations are homogenous, you must constrain the scale of $\mathbf{m}$, so minimize $\mathbf{m}^T \mathcal{D}^T \mathcal{D} \mathbf{m}$ subject to $\mathbf{m}^T \mathbf{m} = 1$. **exercises** The resulting estimate of $\mathcal{M}$ has a good reputation as a start point for a full optimization. It is straightforward to incorporate weights on the points into this estimate. If the weights come from IRLS, then you need this construction only at the start. For every other iteration, the previous iteration will supply an acceptable start point as well as weights.

Things to think about...



15.6. Section 15.3.1 says: " Finding $\mathcal{M}$ now reduces to minimizing

$$\sum_i w_i \left(\mathbf{u}_i - \mathcal{M}\mathbf{v}_i\right)^T \left(\mathbf{u}_i - \mathcal{M}\mathbf{v}_i\right)$$

as a function of $\mathcal{M}$. The natural procedure – take a derivative and set to zero, and obtain a linear system – works fine" What is the linear system you would solve to find $\mathcal{M}$?

15.7. Check that

$$\mathsf{Tr}\left(\mathcal{ABC}\right) = \mathsf{Tr}\left(\mathcal{BCA}\right) = \mathsf{Tr}\left(\mathcal{CAB}\right)$$

for 1×1 matrices; now check this for 2×2 matrices by writing the whole thing out.

15.8. You have a dataset of N points $\mathbf{y}_i$. Write the center of gravity for these points as $\mathbf{c}$. Check the center of gravity of the points $\mathcal{M}\mathbf{y}_i + \mathbf{t}$ is $\mathcal{M}\mathbf{c} + \mathbf{t}$.

15.9. Section 15.4.2 has: "you can apply translations and scales as appropriate to estimate the transformation between the original coordinate systems." Explain.