Using Multiple Images and Contours for Deformable 3D-2D Registration of a Preoperative CT in Laparoscopic Liver Surgery - Supplementary material -

Yamid Espinel, Lilian Calvet, Karim Botros, Emmanuel Buc, Christophe Tilmant, and Adrien Bartoli

Function SVRegistration
  Input: M_0, C_C // Preoperative 3D model, Set of 2D-3D contours
  Output: M // Registered model
  max_iter $\leftarrow$ 100 // Maximum number of iterations
  $\epsilon \leftarrow 10^{-3}$ // Precision threshold
  $M_i \leftarrow M_0$ // Initialise model
  $i \leftarrow 1$ // Current iteration

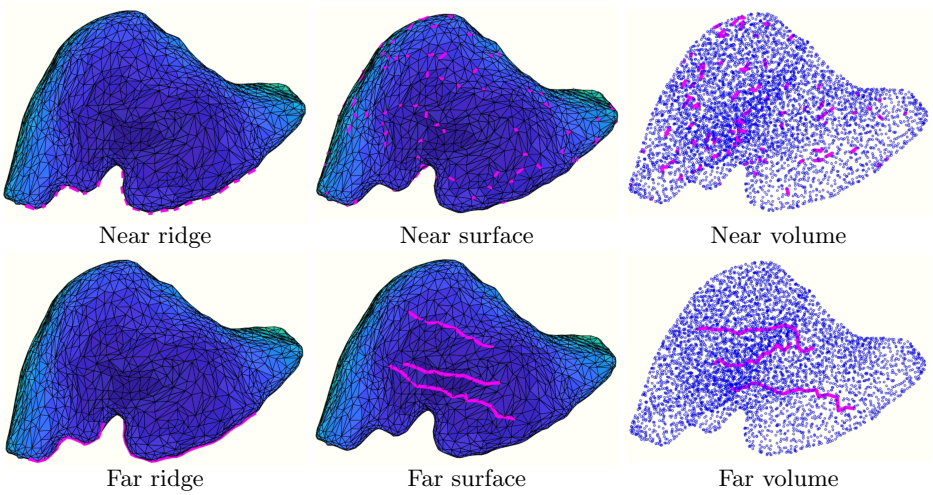
  repeat
    $M_{i+1} \leftarrow$ SolveBiomechanicalConstraints(M_i) // Solve biomechanical constraints
    $M_{i+1} \leftarrow$ SolveContourConstraints(M_{i+1}, C_C) // Solve 3D-2D contour constraints
  until dist$(M_{i+1}, M_i) \leq \epsilon$ ***or*** $i++ \geq$ max_iter

Appendix 1. Algorithm for SV registration.

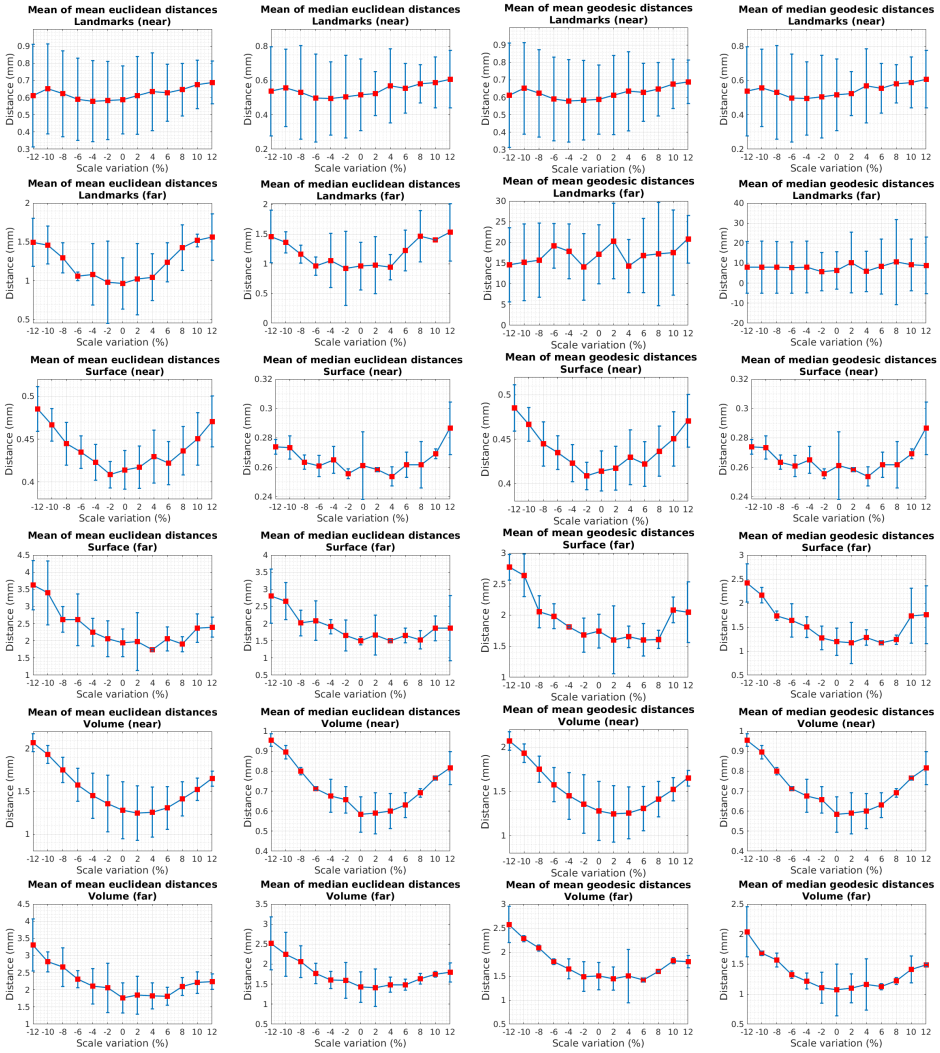
Function MVBRegistration
  Input: $\mathcal{I} = \{I_1, ..., I_N\}, M_0, \mathcal{C}_C = \{C_{C1}, ..., C_{CN}\}$ // Laparoscopic images, Preoperative 3D model, Set of 2D-3D contours
  Output: M // Registered model
  max_iter $\leftarrow$ 100 // Maximum number of iterations
  $\epsilon \leftarrow 10^{-3}$ // Precision threshold
  $M_i \leftarrow M_0$ // Initialise model
  $i \leftarrow 1$ // Current iteration

  repeat
    $n \leftarrow 1$ // View counter
    repeat
      $M_{vn} \leftarrow$ SVRegistration(M_i, C_{Cn}) // Perform single-view registration
    until $n++ \geq$ size($\mathcal{I}$)
    $M_{i+1} \leftarrow$ ComputeAverageModel($\{M_{v1}, ..., M_{vN}\}$) // Average model from all views
    $n \leftarrow 1$ // View counter
    repeat
      $M_{vn} \leftarrow$ RigidICPRegistration(M, C_{Cn}) // Perform rigid ICP registration
    until $n++ \geq$ size($\mathcal{I}$)
  until dist$(M_{i+1}, M_i) \leq \epsilon$ ***or*** $i++ \geq$ max_iter

Appendix 2. Algorithm for MV-B registration.



Appendix 3. Illustration of the measured anisometries in the liver 3D model. The near anisometries are measured in contiguous pairs of points. The far anisometries are measured in non-contiguous pairs of points.



Appendix 4. Behaviors of the 24 anisometry measures for 2 phantoms, using 10 combinations of 8 views per phantom. Two anisometry measures have a convex behavior: the near volume mean euclidean anisometry, and the near volume mean geodesic anisometry.

Function MVB ∗ Registration
  Input: $\mathcal{I} = \{I_1, ..., I_N\}, M_0, \mathcal{C}_C = \{C_{C1}, ..., C_{CN}\}, P, \mathcal{T} = \{T_1, ..., T_N\}$ // Laparoscopic images, Preoperative 3D model, Set of 2D-3D contours, SfM pointcloud, SfM camera poses
  Output: M // Registered model
  max_iter $\leftarrow 20$ // Maximum number of iterations
  $\epsilon \leftarrow 0.01$ // Minimum interval size parameter
  converged $\leftarrow$ false // Binary convergence indicator

  $M \leftarrow$ SVRegistration(M_0, C_{C1}) // Perform single-view registration on the first view
  $s_M \leftarrow$ computeInitialScale(M, P) // Middle SfM scale bound
  $s_L \leftarrow s_M - (0.3 * s_M)$ // Lower scale bound
  $s_U \leftarrow s_M + (0.3 * s_M)$ // Upper scale bound
  $\mathcal{T}_L \leftarrow \mathcal{T} * s_L$ // Scale SfM poses for lower scale
  $\mathcal{T}_U \leftarrow \mathcal{T} * s_U$ // Scale SfM poses for upper scale
  $M_L \leftarrow$ MVRegistration$(\mathcal{I}, M_0, C_C, \mathcal{T}_L,$ max_iter$)$ // Multi-view registration on lower bound
  $M_U \leftarrow$ MVRegistration$(\mathcal{I}, M_0, C_C, \mathcal{T}_U,$ max_iter$)$ // Multi-view registration on upper bound
  $D_L \leftarrow$ ComputeAnisometry(M_0, M_L) // Compute anisometry for lower bound
  $D_U \leftarrow$ ComputeAnisometry(M_0, M_U) // Compute anisometry for upper bound
  repeat
    // If upper bound has a lower anisometry, choose the sub-interval $\{D_M, D_U\}$:
    if $D_L > D_U$ ***and*** $(s_L - s_M) > \epsilon$ **then**
      $D_L \leftarrow D_M$ // Update lower bound anisometry
      $s_L \leftarrow s_M$ // Update lower bound scale
      $s_M \leftarrow (s_M + s_U)/2$ // Compute new middle bound scale
    // If upper bound has a lower anisometry, choose the sub-interval $\{D_L, D_M\}$:
    else if $D_L < D_U$ ***and*** $(s_L - s_M) > \epsilon$ **then**
      $D_U \leftarrow D_M$ // Update upper bound anisometry
      $s_U \leftarrow s_M$ // Update upper bound scale
      $s_M \leftarrow (s_L + s_M)/2$ // Compute new middle bound scale
    // If the difference between bounds is less than a threshold, we have converged:
    else if $(s_L - s_M) \leq \epsilon$ **then**
      $M \leftarrow M_M$ // The resulting model comes from the last middle bound scale
      converged $\leftarrow$ true
    end
    $\mathcal{T}_M \leftarrow \mathcal{T} * s_M$ // Scale SfM poses
    $M_M \leftarrow$ MVRegistration$(\mathcal{I}, M_0, C_C, \mathcal{T}_M,$ max_iter$)$ // Multi-view registration on medium bound
    $D_M \leftarrow$ ComputeAnisometry(M_0, M_M) // Compute anisometry for medium bound
  until converged
Function MVRegistration
  Input: $\mathcal{I}, M, \mathcal{C}_C, P, \mathcal{T},$ max_iter // Laparoscopic images, Preoperative 3D model, Set of 2D-3D contours, SfM pointcloud, SfM camera poses, Maximum number of iterations
  Output: M // Registered model
  $i \leftarrow 1$ // Current iteration

  repeat
    $M \leftarrow$ SolveBiomechanicalConstraints(M) // Biomechanical optimization
    $n \leftarrow 1$ // View counter
    repeat
      $M_{vn} \leftarrow$ SolveContourConstraints(M, C_{Cn}, T_n) // Contour landmarks optimization
    until $n++ \geq$ size$(\mathcal{I})$
    $M_{i+1} \leftarrow$ ComputeAverageModel$(\{M_{v1}, ..., M_{vN}\}, \mathcal{T})$
  until $i++ \geq$ max_iter

Appendix 5. Algorithm for MV-B* registration.

Function MVCRegistration
Input: $\mathcal{I} = \{I_1, ..., I_N\}, M_0, \mathcal{C}_C = \{C_{C1}, ..., C_{CN}\}$ // Laparoscopic images, Preoperative 3D model, Set of 2D-3D contours
Output: M // Registered model
max_iter $\leftarrow 100$ // Maximum number of iterations
$\epsilon \leftarrow 10^{-3}$ // Precision threshold
$M_i \leftarrow M_0$ // Initialise model
$i \leftarrow 1$ // Current iteration

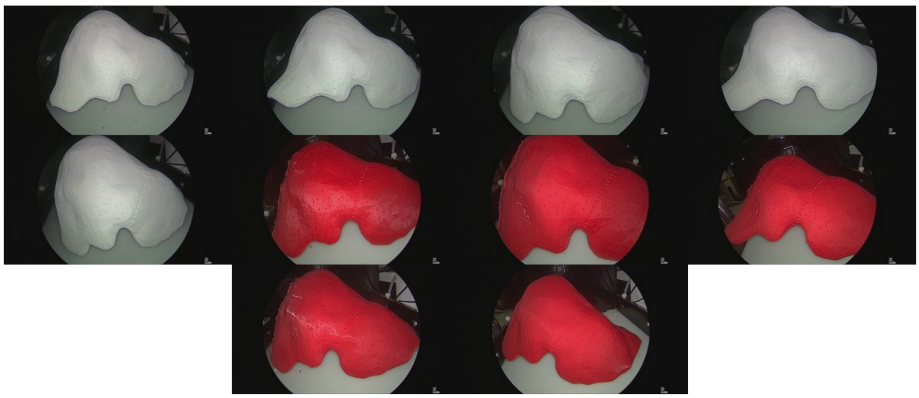
$\{K_C\} \leftarrow$ DetectKeypointCorrespondences($\mathcal{I}$) // Detect inter-image correspondences
$\{\eta\} \leftarrow$ ComputeWarps(K_C) // Compute warp functions
repeat
  $n \leftarrow 1$ // View counter
  repeat
    $M_{vn} \leftarrow$ SVRegistration(M_i, C_{Cn}) // Perform single-view registration
  until $n++ \geq$ size($\mathcal{I}$)
  $M_{i+1} \leftarrow$ ComputeAverageModel($\{M_{v1}, ..., M_{vN}\}$) // Average model from all views
  $n \leftarrow 1$ // View counter
  repeat
    $m \leftarrow 1$ // View counter
    repeat
      $x_{pm} \leftarrow$ ProjectVisibleParticlesIn2D(M_{vm}) // Project model particles in 2D
      $x_{pmn} \leftarrow$ TransferParticlesUsingWarps(x_{pm}, η_{mn}) // Transfer particles to reference view n
    until $m++ \geq$ size($\mathcal{I}$)
    $x_{pn} \leftarrow$ ProjectVisibleParticlesIn2D(M_{vn}) // Project model particles in 2D
    $b_n \leftarrow$ BarycentersFromTransferredParticles($x_{pn}, \{x_{p2n}, ..., x_{pNn}\}$) // Compute barycenters from transferred particles
    $M_{vn} \leftarrow$ RigidICPRegistration(M, C_{Cn}, x_{pn}, b_n) // Perform rigid ICP registration
  until $n++ \geq$ size($\mathcal{I}$)
until dist(M_{i+1}, M_i) $\leq \epsilon$ ***or*** $i++ \geq$ max_iter

Appendix 6. Algorithm for MV-C registration.

Function MVDRegistration
Input: $\mathcal{I} = \{I_1, ..., I_N\}, M_0, \mathcal{C}_C = \{C_{C1}, ..., C_{CN}\}$ // Laparoscopic images, Preoperative 3D model, Set of 2D-3D contours
Output: $\{M_{v1}, ..., M_{vN}\}$ // Registered models
max_iter $\leftarrow$ 100 // Maximum number of iterations
$\epsilon \leftarrow 10^{-3}$ // Precision threshold
$M_i \leftarrow M_0$ // Initialise model
$i \leftarrow 1$ // Current iteration

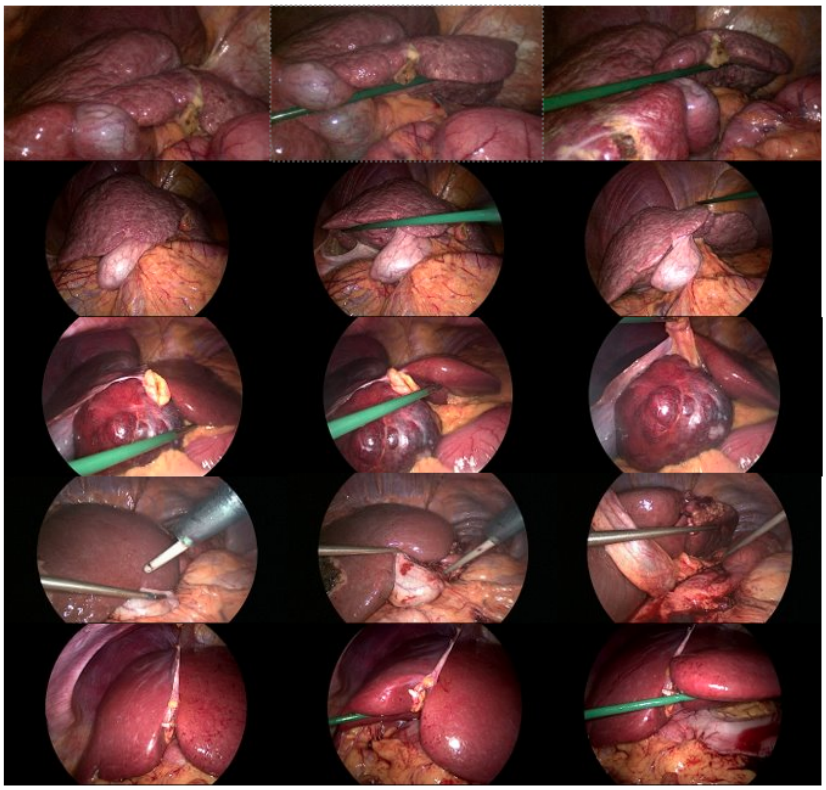
$\{K_C\} \leftarrow$ DetectKeypointCorrespondences($\mathcal{I}$) // Detect inter-image correspondences
$\{\eta\} \leftarrow$ ComputeWarps(K_C) // Compute warp functions
repeat
  $n \leftarrow 1$ // View counter
  repeat
    $M_{vn} \leftarrow$ SVRegistration(M_i, C_{Cn}) // Perform single-view registration
  until $n++ \geq$ size($\mathcal{I}$)
  $n \leftarrow 1$ // View counter
  repeat
    $m \leftarrow 1$ // View counter
    repeat
      $x_{pm} \leftarrow$ ProjectVisibleParticlesIn2D(M_{vm}) // Project model particles in 2D
      $x_{pmn} \leftarrow$ TransferParticlesUsingWarps(x_{pm}, η_{mn}) // Transfer particles to reference view n
    until $m++ \geq$ size($\mathcal{I}$)
    $x_{pn} \leftarrow$ ProjectVisibleParticlesIn2D(M_{vn}) // Project model particles in 2D
    $b_n \leftarrow$ BarycentersFromTransferredParticles($x_{pn}, \{x_{p2n}, ..., x_{pNn}\}$) // Compute barycenters from transferred particles
    $M_{vn} \leftarrow$ RigidICPRegistration($M_{vn}, C_{Cn}, x_{pn}, b_n$) // Perform rigid ICP registration
  until $n++ \geq$ size($\mathcal{I}$)
  $M_{i+1} \leftarrow M_{v1}$
until dist(M_{i+1}, M_i) $\leq \epsilon$ ***or*** $i++ \geq$ max_iter

Appendix 7. Algorithm for MV-D registration.



Appendix 8. Laparoscopic views of the phantoms used in registration. Each view corresponds to a different deformation.

Appendix 9. Example of rigidly-related patient views used in registration. Each row corresponds to a different patient.



Appendix 10. Example of non-rigidly-related patient views used in registration. Each row corresponds to a different patient.