Inductive Aerodynamics

Samuel Wilkinson[1], Sean Hanna[2], Lars Hesselgren[3], Volker Mueller[4]
[1,2]University College London, UK, [3]PLP/Architecture, UK, [4]Bentley Systems, US
[1]smlwilkinson@gmail.com, [2]s.hanna@ucl.ac.uk, [3]lhesselgren@plparchitecture, [4]volker.mueller@bentley.com

Abstract. *A novel approach is presented to predict wind pressure on tall buildings for early-stage generative design exploration and optimisation. The method provides instantaneous surface pressure data, reducing performance feedback time whilst maintaining accuracy. This is achieved through the use of a machine learning algorithm trained on procedurally generated towers and steady-state CFD simulation to evaluate the training set of models. Local shape features are then calculated for every vertex in each model, and a regression function is generated as a mapping between this shape description and wind pressure. We present a background literature review, general approach, and results for a number of cases of increasing complexity.*
Keywords. *Machine learning; CFD; tall buildings; wind loads; procedural modelling.*

INTRODUCTION

It is generally recognised that architects currently require performance information to guide their decisions almost from the inception of a project. In fact, there is a mentality present of simply trying to collect as much data as possible with the intention of synthesising it into a situated design response. This presents a problem, especially for computational fluid dynamic (CFD) wind simulation, whereby the time required to assess the performance is obstructive to the fast and iterative nature of current parametric design softwares. This is possibly due to the tendency for architectural software tools to originate in engineering fields, without due consideration of speed-accuracy tradeoffs to adjust for the application requirements (Chittka et al., 2009; Lu et al., 1991). In other words, they are typically too accurate and slow for the fast pace of modern conceptual design, massing or form decisions. Developing a method that can give real-time performance feedback about a form allows for intuitive play of the kind we are used to with physical models.

Wind engineering has traditionally been within the remit of engineers or specialists, with numerical simulation (CFD) considered a supportive tool to physical boundary layer wind tunnel (BLWT) testing. For instance, in the computational wind engineering (CWE) literature there is substantial caution around numerical analysis, namely for Reynolds-averaged Navier-Stokes (RANS) and to a lesser extent large-eddy simulations (LES) (Stathopoulos, 1997; Bitsuamlak, 2006; Dagnew et al., 2009; Menicovich et al., 2002). However, architects are increasingly getting involved with analysis, where concerns over accuracy are less paramount since demand is typically for relative scenario comparison or general flow behaviour (Lomax et al., 2001; Malkawi et al., 2005; Chronis et al., 2012).

The tall building typology has been identified as a focal area here for a number of reasons. Firstly, as height increases so too do the wind forces (along with seismic and gravitational) which has consequences on facade panelisation and structural efficiency, amongst others. We can construct a simple motivational argument to say that increased external wind force requires more opposing force, i.e. more structure, more materials, larger cores, less letable floor space, less revenue etc. Therefore there is a need to consider the aerodynamic form of these buildings as they increase in height. Secondly, the trend for tall buildings is to build them as high as (contextually, economically and structurally) possible, necessitating cutting-edge design and construction technologies (CTBUH, 2012). Thirdly, tall building form lends itself well to parametric design as there is often a high degree of vertical logic that can be expressed neatly with mathematical expressions (this generalisation is at least more true than for shorter buildings). Given this, it is possible to easily generate a procedural, or generic, tall building model that, with a relatively small number of parameters, can represent a large number of potential designs. This becomes useful when the objective is to sample the typological space of potential buildings, which will be discussed in the methodology.

We present a novel approach to predict wind pressure on tall building models for early-stage generative design exploration and optimisation (exploration as the non-discrete parametric equivalent of tinkering, and optimisation as the single- or multi-objective directed design space search requiring iterative testing and evaluation). The method provides fast surface pressure data with the conventional visualisation, reducing performance feedback time whilst maintaining verisimilitude.

This is achieved through the use of a machine learning algorithm, trained on a pre-computed set of CFD simulation data. *ANSYS CFX 13.0*, a commonly used solver in engineering practice, was used for steady-state RANS with a k-E turbulence model. The learning technique is grouped with artificial neural networks (ANN), support vector machines (SVM), and random forest (RF) decision trees, in that there is a training set of cases from which generalised rules are generated (Duffy, 1997). The term machine learning stems from the fields of computer science (Mitchell, 1997) and artificial intelligence (Samuel, 1959), but in statistics is referred to as regression and in engineering as function approximation or surrogate modelling. Once trained, this enables us to provide a new test case and make a prediction of the outcome. Inductive reasoning, epistemologically, means constructing generalisations from specific information, as opposed to deductive reasoning where small details are construed from generalisations. The fundamental outcome of this learning approach is therefore a continuous output response allowing interpolation and extrapolation between cases that have not been explicitly simulated. In doing so, we are essentially moving the simulation time from the front-end to the back-end of the process where more time is available for precomputation.

The following section provides a review of relevant literature in the generative, performative design of tall buildings, wind modelling methods, speed-accuracy tradeoffs, incorporation of learning in design, concluding with a problem-solution hypothetical argument positioned in this state of current literature. The subsequent structure of this paper will describe the methodological approach in general terms, and results are presented from a series of experimental case studies of increasing complexity from trivial to practical. The conclusions, further work and the paper as a whole are positioned within the scope of ongoing research.

LITERATURE REVIEW

Tall Buildings

Tamura et al. (2009; 2010) and Tanaka et al. (2012) acknowledge the increase in tall building complexity beyond the traditional extruded rectilinear form. We are now seeing more unconventional free-style forms derived from the architect's use of more advanced modelling software. These new complicated sectional shapes that may vary with height, can actually provide better aerodynamic performance by disrupting, or 'confusing', vortex shedding and thus reducing crosswind response. Benefits can also be found in more subtle manipulations such as corner chamfering or cutting, and by creating voids, or porous regions, near the edges.

Despite rapid advances over the past century, this emerging generation of skyscrapers poses new challenges for wind engineering. Irwin (2009) discusses a number of these, such as the impact that aerodynamics have on construction cost. Since the structure itself is a large proportion of the cost, and as for tall buildings the wind is the governing lateral load, there are significant benefits to be had from reducing wind loads. This also has the effect of reducing lateral motions that can potentially cause occupant discomfort. He also suggests that shape aerodynamics must be proactively considered, and iteratively optimised, early on in the design. With the new generation of super-tall towers over 600m it is simply not possible to ignore the wind performance. He quotes a designer of the Burj Khalif, saying *"we practically designed the tower in the wind tunnel"*, and were therefore able to produce an extremely efficient aerodynamic shape that enabled the height with reasonable structural systems and costs, and without any damping system.

The increase in the use of parametric CAD softwares has seen a rise in the last decade namely with the release of *Bentley GenerativeComponents* and *Rhino Grasshopper*, plus more generally with the increased adoption of scripting. These allow the user to create parametrically associative relationships related to geometry. The extension of this idea is to use rules to define the parameters, or where these rules can be related to the performance of a model component the geometry is directed by some evaluative metric. Certain metrics can be calculated quickly without problem, but if the calculation takes time it becomes obstructive to the modelling process. We adopt the premise that it is better to have a broader range of lower resolution data rather than a limited amount of exact data.

Speed-Accuracy Tradeoffs

Speed-accuracy tradeoffs (SATs) show that response accuracy generally increases with response time, i.e. taking more time to make a decision results in a better decision. Biological examples have been noted by Chittka et al. (2009), who explains that *"when it takes a long time to solve a difficult task, and the potential costs of errors are low, the best solution from the perspective of an animal might be to guess the solution quickly, a strategy that is likely to result in low decision accuracy."* The two extremes can be called impulsive and reflective. This provides a neat analogy for performance analysis in design where it is necessary to consider what the application of the simulation tool is, and the consequent risks, before deciding a suitable accuracy.

Crucially though, and in conjunction with this reasoning, Burns (2005) demonstrates that making more decisions with more mistakes (fast and inaccurate) results in better overall performance (with bees, more nectar collected) than the more fastidious (slow and accurate). Defining accuracy as the proportion of choices that are correct, this highlights that accuracy should not be confined to the immediate task, i.e. simulation accuracy, but to the larger one of improving building performance (Figure 1).

Response time is critical for performance-driven design and SATs must be considered when developing early stage tools for when large-scale decisions are made. Performance information is often scarce at this stage and iterative decisions must be made quickly, necessitating fast response times in sync with the project cycles. The development of CFD models have been focused over the past decades on improving accuracy, and computational time is optimised by specific software vendors after-the-fact, with little thought given to the accuracy required by the user. In contrast, recent developments in computer graphics have started with the desired accuracy (believable) and speed (real-time) in mind, with successful results.

In the design context, CFD can typically be used for a number of purposes: analysis of internal air movement, pollution dispersion, noise propagation, pedestrian comfort in urban environments or tall building aerodynamics. As mentioned previously, it is the last that is the focal application here, especially for early design stages. There is a paradox here,

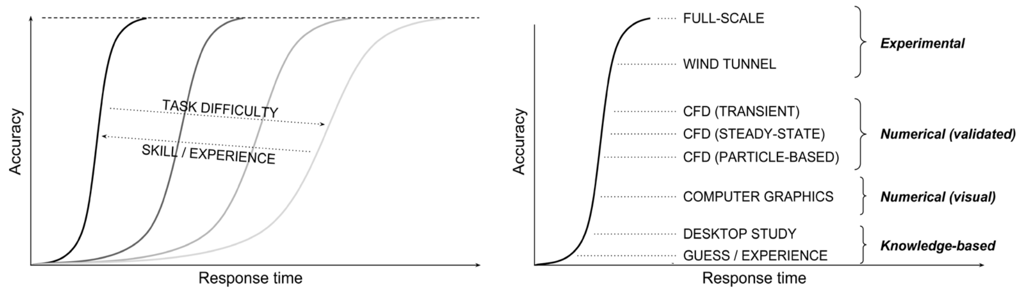

Figure 1
(Left) SAT for various task difficulties and skills; (Right) Notional positions of different modelling methods on SAT.

in that the most complex flow types (bluff bodies) and therefore most computationally intensive, need to be modelled in a scenario where fast results are required. The numerical method must be as accurate and fast as possible. In fact, the conclusion is reached that the fastest method has poor accuracy and the slowest the best accuracy (as would be expected, considering the speed-accuracy tradeoffs mentioned earlier). There is general agreement between (Lomax et al., 2001) and (Chronis et al., 2012) that the *"level of accuracy of a CFD simulation needs to be compromised with the turnaround time requirements of its application."*

Lu et al. (1991) describe the same issue in mechanical engineering where slow but accurate simulation makes interactive decision making impossible, when only quick estimates are desired at early stages. It is only towards the final stages of design, *"when the engineer has converged to a small region of decision space, more accurate simulations are needed to make fine distinctions."* The problem has therefore been present since the early 90s, but as a solution they propose integration of simulation, optimisation and machine learning.

Inductive Learning in Design

Our approach is supported by Samarasinghe (2007), who identifies the best solution to predicting system behaviour through observational data. This is necessary when there is little or no understanding of the *"underlying mechanisms because of complex and non-linear interactions among various aspects of the problem."* Extracting these complex relationships is often difficult since the systems are typically natural, and therefore can have randomness, heterogeneity, multiple causes and effects, and noise. Even when they are successfully extracted, they may be beyond our understanding and are held as intractable computational functions or data structures. Hanna (2011) tests the hypothesis that it is unnecessary to have any understanding of this underlying system behaviour, but rather it is possible to make predictions about the system simply by making observations. This is demonstrated by learning the structural behaviour of system components and applying them to larger-scale scenarios.

Graening et al. (2008) propose a method that allows the extraction of comprehensible knowledge from aerodynamic design data (jet-blades) represented by discrete unstructured surface meshes. They use a displacement measure in order to investigate local differences between designs and the resulting performance variation. Knowledge, or rule, extraction from CFD data is primarily used to guide human- centred design by improving understanding of the system's behaviour, whether it is for jet turbine blade optimisation or architectural design. Whilst the connection between local geometric features and surface pressure has been extended and changed here, and used for a different application, this work is a close precedent.

Problem Hypothesis

It is argued here that approximations of CFD simulations can be made with machine learning regression, using geometric shape descriptors as the learn-

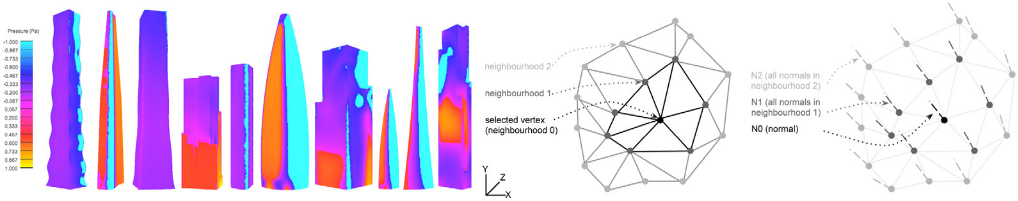

Figure 2
(Left) Examples of evaluated procedural models in the training set on Case 4; (Right) Mesh feature extraction.

ing features. The entire evaluation process can be broadly split into five key work areas: i) procedural geometry generation; ii) batch simulation; iii) shape feature generation; iv) machine learning training; v) prediction and visualisation. Feature generation is essentially the core of the process since the solution depends heavily on geometric description so as to define surface pressure as a function of it. We hypothesise that surface pressure distribution arising from wind flow around tall buildings can be learnt and predicted with an accuracy appropriate to early stage design (feedback from practice indicates <20% error) using shape feature description. It can be shown that it is possible to combine, with an acceptable error, methods that have the separate contradictory objectives of predictive accuracy and speed.

METHODOLOGY

Data Set Generation: Procedural Modelling

The parametric model was created in *Bentley GenerativeComponents*. The goal was to create a generalised tower model, with the two properties of minimising the number of parameters used whilst maximising the design representation potential, i.e. the number of possible buildings it could create. This is important when considering optimisation or exploratory design space searches to avoid the curse of dimensionality. This means that as the number of variables increases, the design space increases exponentially by n^D, where n is the number of samples taken per parameter and D is the number of parameters, or dimensionality. There is therefore clearly a compromise to be made between model efficiency and representability.

The geometry for the training set was generated using a procedural tall building model with a select number of key parameters (Figure 2). There are in fact three separate topologies in the procedural model with their own parameters, since it is difficult to incorporate the entire design space with one parametric logic (Park et al., 2004; Samareh, 1999). Using the unstructured triangulated surface mesh from these means we are not limited by a single parametric topology in the learning phase of the method (Graening et al., 2008). Local surface-mesh shape characteristics are used as input features to the learning algorithm instead of the design parameters, avoiding reliance on any one parametric model definition.

Simulation Method

An established solver, *ANSYS CFX* 13.0, was used throughout to run the RANS steady-state simulations, with a k-ε turbulence model as it is regarded as the most robust. Each simulation, depending on the complexity, requires up to 60 minutes to converge (on a 2.66GHz i7). Solver convergence is reached when residuals fall below a minimum of 1^{-6}, typically at around 100 to 200 iterations. The number of cells in the tetrahedral meshes varies between $0.8\text{x}10^6$ and $1.5\text{x}10^6$ depending on the geometry, with prismatic expansion on surfaces 3 cells deep and a minimum cell size of 0.1m. The wind was applied at an upstream inlet, with a reference speed (U_r) of $1ms^{-1}$ at a reference height (Z_r) of 10m. The most commonly used distribution of mean wind speed with height is the 'power-law' expression:

$$U_x = U_r (Z_x / Z_r)^{\alpha} \qquad (1)$$

The exponent α is an empirically derived coefficient that is dependent on the stability of the atmosphere. For neutral stability conditions it is approximately 0.143, and is appropriate for open-surroundings such as open water or landscape. Future work will include a wind profile that takes surrounding surface roughness, or context, into account, as well as potential wind direction change with height.

Shape Features and Learning

This method creates a definition for the pressure at a point on the model as the function of a local geometric description. To describe a simple example of the process: there are N models of a cuboid with various orientations; each is evaluated, and the pressure P is extracted at M points over each model; for every M, a shape descriptor X is calculated, such as the vertex height, normal components, curvature, etc; this gives a set of geometric characteristics, and a corresponding pressure value; these sets of P(X) are used as the training data. Pressure distribution is predicted from these geometric descriptors alone meaning the selection is critical. A sensitivity analysis has been conducted with a variety of descriptors to determine suitable representation, details of which are not included here. When a new case is presented, the shape descriptors are calculated and used to make a prediction of P. The feature definition for point pressure in $\mathbf{R}^{22}$ vector space used throughout the following is:

$$P\,(Z, N_{(x,y,z)}, N\sigma^{1\text{-}5}{}_{(x,y,z)}, U_{(x,y,z)}) \qquad (2)$$

For a specific model vertex, P is the surface pressure, Z is the height, $N_{(x,y,z)}$ are the normal components, $N\sigma^{1\text{-}5}{}_{(x,y,z)}$ is the standard deviation σ of normal components of cumulative mesh neighbourhood rings 1 through 5, and $U_{(x,y,z)}$ are the normalised model position components. The extent of the neighbourhood curvature can be extended beyond 5 rings, within computational resource limits. The definition in Equation 2 gives 22 inputs and 1 output feature to train the learning algorithm for all cases described below.

For the Orientation, Height and Topology cases, an Artificial Neural Network (ANN) was used, with a 70:30% split of the provided data to training:validation. For the first two cases, separate sets constituting entire models were also held back for testing, i.e. training was at 15° and 20m intervals respectively. For the third case, there was no extra test set but the whole was split 70:15:15% to training:validation:test. Validation data is to check for convergence during training. For the fourth case, training data was from the procedural tall building model and test data from another set of real buildings. In this case, a Random Forest (RF) algorithm was used instead as it provided better results for the more complex problem. Further work is needed with both methods to understand their applicability to certain tasks, however it is known that the RF is better with noisy data sets than the ANN. Training set sizes and summary results are given in Table 1, and computation times are given in Table 2.

RESULTS

Cuboid Orientation

The first and most simple test is the rotation of a cuboid, of width and depth 10m, and height 50m. Simulations were run at 5° intervals from 0 to 85°, and the ANN trained on 15° and tested at 5° intervals. The sensitivity analysis here varies the number of training samples and measures the standard deviation, σ, of the difference between simulation and prediction. Figure 3 (left) shows the error σ against orientation for various set sizes (bold vertical lines are training intervals of 15°), (centre) the training regression of the entire set, and (right) the prediction error for an orientation of 25°. With less training data, it can be seen that error is highest around 45° when flow bifurcations (regime change) occur, although this is negated with sufficient data.

Cuboid Height

Secondly, a parametric cuboid was created with width and depth 10m, and height varying from 10 to 100m in 5m increments. Figure 4 (left) shows the variability when trained on 10, 20, 30 and 45m intervals, and (right) the prediction error for a height of 25m when trained at 20m intervals.

Table 1
Summary of minimum and maximum error standard deviations (% over test case pressure range).

Case	Min σ Error (%)	Max σ Error (%)	Training Set Size
Orientation	1.2 (55°)	1.6 (10°)	110000 (15° training intervals)
Height	0.7 (10m)	2.0 (50m)	44720 (20m training intervals)
Topology	1.8 (5 Edges)	3.5 (0 Edges)	50000
Real	4.8 (Bank of China)	18.3 (Euston)	100000 (Procedural training)

Table 2
*Summary of time (seconds) required for each case, split into Training (one-off back-end time) and Prediction (front-end time). Mean feature generation time is 0.085s/vertex. *Mean over all test set. †After down-sampling.*

Case	Train Sim.	Train Feat. Gen. †	Train	Predict Feat. Gen. *	Predict *
Orientation	21600	9060	2600	1540	< 0.1
Height	18000	2370	720	620	< 0.1
Topology	32400	4670	1060	1750	< 0.1
Real	2160000	12000	620	720	< 0.1

Topology

Here the number of edges was varied from 3 to 10, with 0 (circle), diameter 10m and height 50m. Instead of keeping a complete model separate for testing as in the last two cases, here all cases were used but only a fraction of the total data set was used. This is varied in Figure 5 (left), with a training set ranging from 10000 to 50000.

Tall Buildings

In the final case, training data was collected from simulations of 600 procedural tall building models, with a total of over 4×10^6 shape features extracted. This was down-sampled to 10^5 by removing features in close proximity to reduce training time. The test set contains 10 real tall buildings from around the world, selected for their range of unique architectur-

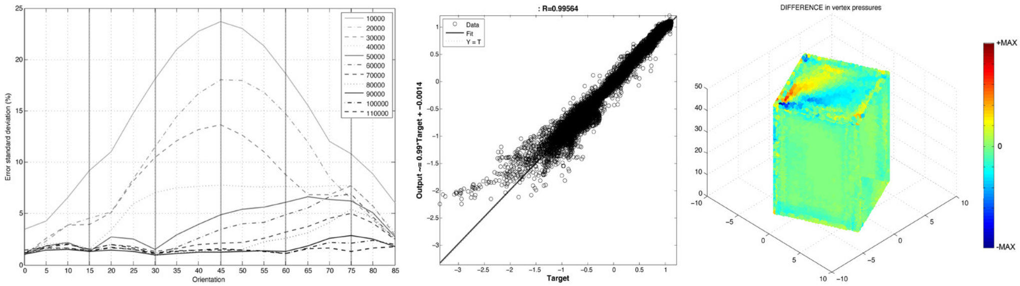

Figure 3
(Left) Orientation vs. Error σ %; (Centre) Training set regression, R=0.99564; (Right) Prediction error (25°).

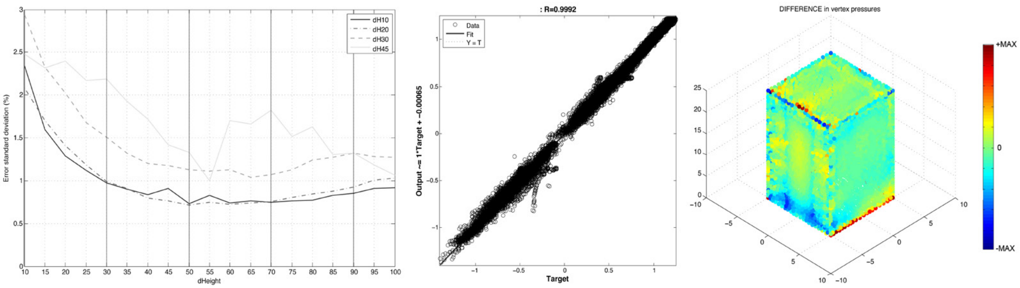

Figure 4
(Left) dHeight vs. Error σ %; (Centre) Training set regression, R=0.9992; (Right) Prediction error (25m).

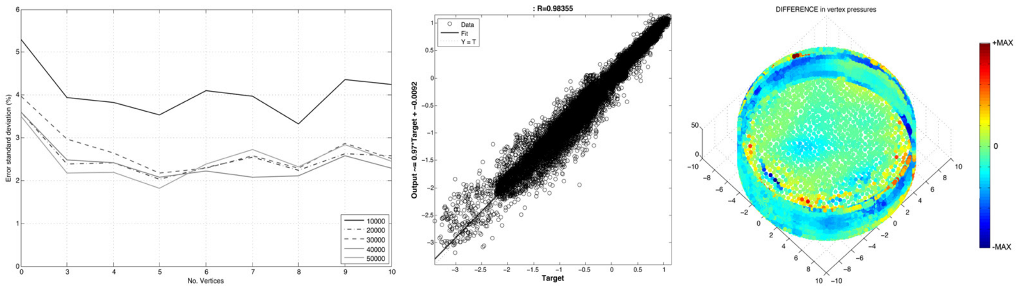

Figure 5
(Left) No. Edges vs. Error σ %; (Centre) Training set regression, R=0.98355; (Right) Prediction error (n0).

al characteristics. Figure 6 shows predicted surface pressure distribution in the top row, and the error distribution for the set in the bottom row. The pressure range (-5.5 to 2.0 Pa) was taken over the entire test set, as was the absolute error range (0 to 65.2%). The error distribution is shown in Figure 7 (right), which fits a Gaussian normal distribution. Error percentiles: 99^{th} = 35.7%, 95^{th} = 20.0%, 90^{th} = 13.0%, 75^{th} = 6.1%. That is, 75% of the test features have an error below 6.1%.

CONCLUSION

The results show that it is possible to achieve a relatively small prediction error (Figure 7 and Table 1) for less time (Table 2), with the methodology and constraints described. These prediction errors are necessary for the compromise in avoiding considerably intensive CFD simulation. Traditionally, for every individual CFD simulation the process can take a minimum of 1 hour, compared to our methodology that has a total front-end prediction time of under 12 minutes (for feature generation and prediction) and a back-end, one-off training set simulation time of 600 hours (for the real case). Once trained, an unlimited number of predictions can then be made.

Whilst these preliminary results are outside the rigorous accuracy necessary for final engineering analysis, they are within the boundaries acceptable

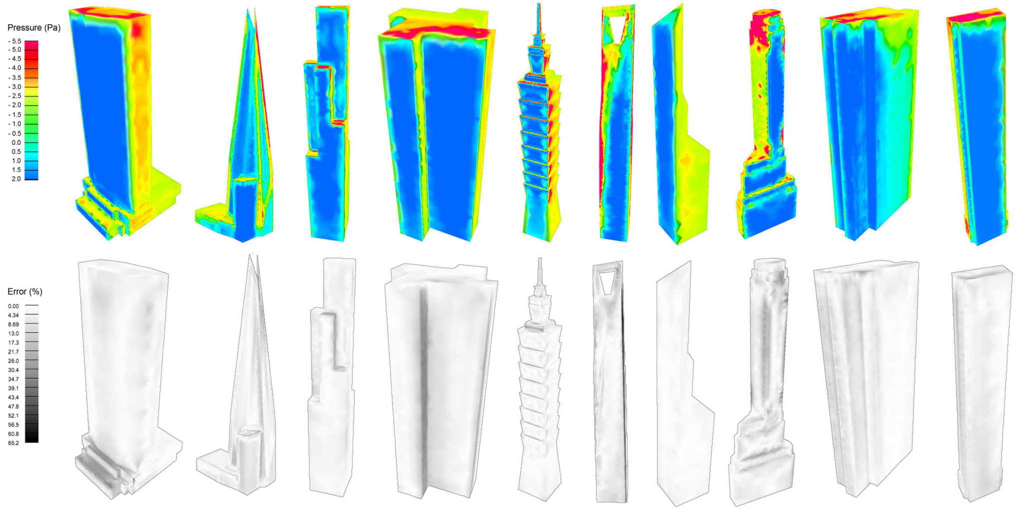

Figure 6
(Upper) Predicted pressure, Pa; (Lower) Error, %. Pressure range is the min. and max. of the entire set for comparison, the error range is absolute max. error of the set (65.2%). (Left to right) (1) Metlife Building, NYC; (2) The Shard, London; (3) Willis Tower (Sears), Chicago; (4) Euston Tower, London; (5) Taipei 101, Taiwan; (6) Shanghai World Financial Centre; (7) Bank of China; (8) Exchange Place, NYC; (9) Frankfurter Buro Centre, Frankfurt; (10) Washington Street, NYC.

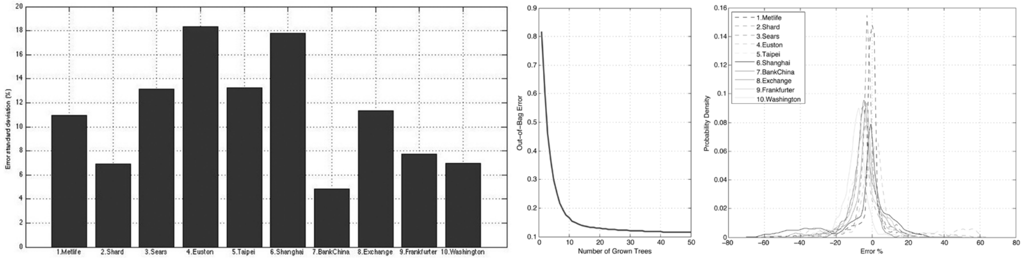

Figure 7
(Left) Error σ % for each case; (Centre) Random Forest learning convergence; (Right) Error probability density.

for early-stage concept design for tall buildings, where interactive response time is a significant consideration. The prediction accuracy and response times achieved are promising for further work given the well-known complexities of fluid behaviour.

The next stages of the work are to consider time-dependent simulations to fully consider the approximation of turbulence, vortex shedding and gusts, as well as interference from complex urban contexts on boundary conditions, and further improvement to the shape feature selection and generation time.

ACKNOWLEDGEMENTS

This research was sponsored by the EPSRC, Bentley Systems and PLP Architects.

REFERENCES

Bitsuamlak, G., 2006. *Application of computational wind engineering: A practical perspective.* In Third National Conference in Wind Engineering. pp. 1–19.

Burns, J.G., 2005. *Impulsive bees forage better: the advantage of quick, sometimes inaccurate foraging decisions.* Animal Behaviour, 70(6), pp.1–5.

Chittka, L., Skorupski, P. & Raine, N.E., 2009. *Speed-accuracy tradeoffs in animal decision making.* Trends in ecology & evolution, 24(7), pp.400–7.

Chronis, A. et al., 2012. *Design Systems, Ecology and Time.* In ACADIA.

CTBUH, 2012. *Tall Buildings in Numbers : A Tall Building Review,* 2012(1).

Dagnew, A.K., Bitsuamlak, G. & Merrick, R., 2009. *Computational evaluation of wind pressures on tall buildings.* In 11th Americas Conference on Wind Engineering.

Duffy, A.H.B., 1997. *The "what" and "how" of learning in design.* IEEE Expert, 12(3), pp.71–76.

Graening, L. et al., 2008. *Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries.* JMMA, 7(4), pp.329–350.

Hanna, S., 2011. *Addressing complex design problems through inductive learning.*

Irwin, P.A., 2009. *Wind engineering challenges of the new generation of super-tall buildings.* JWEIA, 97(7-8), pp.328–334.

Lomax, H., Pulliam, T.H. & Zingg, D.W., 2001. *Fundamentals of computational fluid dynamics,* Berlin: Springer.

Lu, S.C.-Y., Tcheng, D.K. & Yerramareddy, S., 1991. *Integration of Simulation, Learning and Optimization to Support Engineering Design.* Annals of the CIRP, 40(1), pp.143–146.

Malkawi, A.M. et al., 2005. *Decision support and design evolution: integrating genetic algorithms,* CFD and visualization. AIC, 14(1), pp.33–44.

Menicovich, D. et al., 2002. *Generation and Integration of an Aerodynamic Performance Database within the Concept Design Phase of Tall Buildings.*

Mitchell, T.M., 1997. *Machine Learning,* McGraw-Hill.

Park, S.M. et al., 2004. *Tall Building Form Generation by Parametric Design Process.* In CTBUH. Seoul Conference, pp. 1–7.

Samarasinghe, S., 2007. *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition,* Auerbach Publications, NY.

Samareh, J.A., 1999. *A Survey of Shape Parameterisation Techniques.* In CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics. pp. 333–343.

Samuel, A.L., 1959. *Some Studies in Machine Learning Using the Game of Checkers.* IBM JRD, 3(3).

Stathopoulos, T., 1997. *Computational wind engineering:*

Past achievements and future challenges. JWEIA, 67-68, pp.509–532.
Tamura, Y., 2009. *Wind and tall buildings.* In EACWE 5.
Tamura, Y. et al., 2010. *Aerodynamic Characteristics of Tall Building Models with Various Unconventional Configurations.* Structures Congress 2010, pp.278–278.
Tanaka, H. et al., 2012. *Experimental investigation of aerodynamic forces and wind pressures acting on tall buildings with various unconventional configurations.* JWEIA, 107-108, pp.179–191.